PESARO 2012

The Second International Conference on Performance, Safety and Robustness in Complex Systems and Applications

ISBN: 978-1-61208-198-4

April 29 - May 4, 2012

Chamonix / Mont Blanc, France

PESARO 2012 Editors

Petre Dini, Concordia University, Canada / China Space Agency Center, China

Eugen Borcoci, Politehnica University of Bucharest, Romania

PESARO 2012

Foreword

The Second International Conference on Performance, Safety and Robustness in Complex Systems and Applications [PESARO 2012], held between April 29th and May 4th, 2012 in Chamonix / Mont Blanc, France, continued the inaugural event dedicated to fundamentals, techniques and experiments to specify, design, and deploy systems and applications under given constraints on performance, safety and robustness.

There is a relation between organizational, design and operational complexity of organization and systems and the degree of robustness and safety under given performance metrics. More complex systems and applications might not be necessarily more profitable, but are less robust. There are trade-offs involved in designing and deploying distributed systems. Some designing technologies have a positive influence on safety and robustness, even operational performance is not optimized. Under constantly changing system infrastructure and user behaviors and needs, there is a challenge in designing complex systems and applications with a required level of performance, safety and robustness.

We take here the opportunity to warmly thank all the members of the PESARO 2012 Technical Program Committee. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to PESARO 2012. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the PESARO 2012 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that PESARO 2012 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of performance, safety and robustness in complex systems and applications.

We are convinced that the participants found the event useful and communications very open. We also hope the attendees enjoyed their stay in the French Alps.

PESARO Advisory Committee:

Piotr Zwierzykowski, Poznan University of Technology, Poland
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Yulei Wu, Chinese Academy of Sciences, China
Harold Liu, IBM Research, China

PESARO 2012 PROGRAM COMMITTEE

PESARO Advisory Committee

Piotr Zwierzykowski, Poznan University of Technology, Poland
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Yulei Wu, Chinese Academy of Sciences, China
Harold Liu, IBM Research, China

PESARO 2012 Technical Program Committee

Salimur Choudhury, Queen's University - Kingston, Canada
Juan Antonio Cordero, École Polytechnique / INRIA, France
John-Austen Francisco, Rutgers University - Piscataway, USA
Mina Giurguis, Texas State University - San Marcos, USA
Mesut Günes FU-Berlin, USA
Charles Kamhoua Kenmogne, Air Force Research Laboratory, USA
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Harold Liu, IBM Research, China
Kia Makki, Technological University of America (TUA), USA
Birgit Milius, Institut fuer Eisenbahnwesen und Verkehrssicherung (IfEV) /Technische Universitaet – Braunschweig, Germany
Liam Murphy, University College Dublin, Ireland
Asoke K. Nandi, The University of Liverpool, UK / University of Jyvaskyla, Finland / University of Calgary, Canada
Harald Øverby, NTNU, Norway
Maciej Piechowiak, Kazimierz Wielki University - Bydgoszcz, Poland
M. Zubair Rafique, King Saud University - Islamabad, Pakistan
Roger S. Rivett, Land Rover - Gaydon, UK
Dhananjay Singh, Electronics and Telecommunications Research Institute (ETRI), South Korea
Mukesh Singhal, University of Kentucky, USA
Young-Joo Suh, Pohang University of Science and Technology (POSTECH), South Korea
Zhi Sun, Georgia Institute of Technology, USA
Batool Talha, University of Minnesota, USA
Yuejin Tan, National University of Defense and Technology - Changsha, China
Yang Wang, Georgia State University, USA
Yun Wang, Bradley University - Peoria, USA
Jun Wu, National University of Defense and Technology, China
Yanwei Wu, Western Oregon University, USA
Yulei Wu, Chinese Academy of Sciences, China
Gaoxi Xiao, Nanyang Technological University, Singapore
Bin Xie, InfoBeyond Technology LLC - Louisville, USA
Bashir Yahya, University of Versailles, France
Nabila Zbiri, Université d'Evry Val d'Essonne, France
Yanmin Zhu, Shanghai Jiao Tong University, China
Piotr Zwierzykowski, Poznan University of Technology, Poland

Copyright Information

Table of Contents

Execution Management of Applications with Runtime Restrictions on Opportunistic Grids Environments

Marcio Rodrigo Melo Martins, Berto de Tácio Pereira Gomes, Jesseildo Figueredo Gonçalves
Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia de Eletricidade
Av. dos Portugueses s/n, Campus Universitário do Bacanga, CEP. 65085-580, São Luís-MA-Brasil
marciorodrigomm, bertodetacio, jesseildo@gmail.com

Francisco José da Silva e Silva
Universidade Federal do Maranhão, Departamento de Informática
fssilva@deinf.ufma.br

***Abstract*—An opportunistic grid computing environment takes advantage of idle computing cycles of regular computers and workstations that can be spread across several administrative domains for running high performance applications. Opportunistic grids are usually constructed from personal computers that do not need to be dedicated for executing grid applications. The grid workload must coexist with local applications executions, submitted by the nodes regular users. Thus, its execution environment is typically dynamic, heterogeneous and unpredictable failures occur frequently. In addition, the resources of an opportunistic grid can be used at any time for the execution of local tasks, making it difficult to preview the conclusion of the tasks running on the grid nodes. These characteristics hinder the successful execution of applications for which there are time restrictions related to its completion. This paper presents a management mechanism specifically designed for opportunistic grid computing environments for handling the execution of applications with time deadlines set by users during their submission to the system. The proposed mechanism is based on a dynamic scheduling and rescheduling approach and was evaluated using a simulated model considering various typical scenarios of opportunistic grids. The results demonstrated the benefits of the proposed approach in comparison to traditional scheduling approaches applied in opportunistic grids.**

***Keywords*-opportunistic grids; application scheduling; soft deadline; simulation.**

I. Introduction

A computer grid is a computing system that coordinates distributed resources using standard protocols and interfaces to enable integration and sharing of computational resources, such as computing power, software, peripherals and data on corporate networks and between institutions. The computer grid technology currently receives great attention from both academia and industry since it have established itself as an attractive alternative for executing a wide range of applications that require massive computational power or process large volumes of data in various areas, such as computational biology, weather and market simulations.

Currently, corporations and universities typically have hundreds or thousands of desktop machines, which are used by workers as their personal workstations or by students in instructional and research laboratories. When analyzing the usage of each of these machines one concludes that they sit idle for a significant amount of time. Even when the computer is in use, it normally has a large portion of idle resources. Opportunistic grid middleware enables the use of the existing computing infrastructure available in laboratories and offices in universities, research institutes, and companies to execute computationally intensive applications. They are usually constructed from personal computers that do not need to be dedicated for executing grid applications. The grid workload must coexist with local applications executions, submitted by the nodes regular users.

The process of application scheduling in a grid system consists in assigning the applications tasks to available resources in accordance to specific goals, such as to minimize the applications response time and/or maximize the use of computational resources. However, the construction of a scheduling strategy focused on opportunistic grid environments is a challenging task due to several features present in these computing environments, such as: (a) instability, that arises from the fact that nodes are not dedicated and applications do not run in a controlled environment; (b) high heterogeneity of computing nodes and network links, usually comprising resources spread across different administrative domains; (c) the middleware goal of not interfering with the regular use of resources, which requires migration and rescheduling of applications when resources become unavailable. In addition, the resources of an opportunistic grid can be used at any time for the execution of local tasks, making it difficult to preview the conclusion of the tasks running on the grid nodes. In particular, these characteristics hinder the successful execution of applications for which there are time restrictions related to its completion.

This paper presents a management mechanism specifically designed for opportunistic grid computing environments for handling the execution of applications with time deadlines set by users during their submission to the system. The proposed mechanism is based on a dynamic scheduling and rescheduling approach and was evaluated using a simulated model considering various typical scenarios of opportunistic grids. The results demonstrated the benefits of the proposed approach in comparison to traditional scheduling approaches applied in those environments. The text is structured as follows: Section II describes the related work that influenced the development of this work. Section III presents our proposed mechanism, its concepts, components, and implementation. Section IV describes the results of several experiments that were conducted for evaluating the proposed approach. Finally, in Section V, we describe the conclusions derived from this work and presents future perspectives arising from this initial effort.

II. Related Work

Currently, there are several works related to the development of scheduling algorithms for computer grids aiming different performance goals, such as minimizing the completion time of application or to maximize the use of grid resources, without taking into account the users requirements [1] [2] [3] [4]. Other studies have also emerged with the purpose of meeting user defined Quality of Service (QoS) metrics [5] [6] [7]. These works emphasizes the difficulties of providing support for execution time constraints of submitted applications (*deadlines*). They present new approaches to scheduling, or the adequacy of existing approaches, in order to consider QoS requirements and metrics for the evaluation of scheduling strategies and/or fault tolerance mechanisms in computer grids.

Buyya et al. [5] present a scheduling algorithm for Parameter Sweep applications on global grids. The algorithm goal is to map applications to resources considering, whenever possible, the best cost-benefit ratio between the applications execution time and the cost to perform these computations. Thus, if multiple machines offer the same completion time for a given application, it should be scheduled to the one that offers the least cost (QoS requirement). The algorithm is called *DBC cost-time optimization*. The grid user optionally provides the following parameters when submitting an application for execution: the time she/he is willing to wait until the completion of the application execution (deadline), and the price she/he is willing to pay for the realization of the computation (budget). The algorithm follows an on-line approach. It runs until there are applications to be scheduled able to be executed according to the provided deadline and cost, as defined by its users. The algorithm was evaluated using the GridSim simulator.

Yu and Buyya [6] present a genetic algorithm for scheduling workflow applications. The purpose of this algorithm is to run applications within a certain period with the lowest possible cost (QoS). In order to evaluate the proposed approach, the authors implemented the algorithm and compared it with a set of non-genetic heuristics for two different types of workflow applications (balanced and unbalanced) using the GridSim simulator.

Kyong Kim et al. [7] address an issue that is becoming increasingly important in cluster computing environments: the need to reduce energy consumption for running applications, which gave rise to the term power-aware scheduling. The authors present two scheduling algorithms for Bag-of-Tasks applications, a shared space policy resource allocation, and a time-shared based one. The shared space policy allows the execution of only one task at a time in a given processing unit, while the time-shared policy allows multiple tasks to share the same processing unit, running in alternating time slices. The algorithms goal is to meet time constraints set for the execution of applications, minimizing the energy consumption in cluster systems. An adopted assumption is that the grid nodes have support to Dynamic Voltage Scaling (DVS), a power management technique that allows the dynamic adjustment of the voltage used by a given hardware component. According to the authors, the proposed algorithms can reduce energy consumption by adjusting the levels of voltage used by the grid nodes.

What distinguishes our work from the above described approaches is the specific attention to opportunistic grid environments. In this way, we consider the possible existence of local workload on the grid nodes, on which we have no control. This prevents the accurate prediction of the tasks execution time, forcing the use of a mechanism to monitor the tasks execution progress and the possible need for rescheduling and migrating them. Moreover, we also take into consideration that desktop grids are subject to various types of failures and may exhibit physical problems (from both, nodes and network links), logical errors (in the application or the communication protocols, for example) and suffer invasions of malicious software. Another common failure type is related to the resources dynamism, which are usually not dedicated and can suddenly become unavailable, even when performing computations on behalf of the grid. In addition, applications are usually composed of long-running tasks, which can take hours or even days to run, exacerbating the possibility of failure [8]. In order to circumvent that, we integrated to our solution an application execution autonomic fault tolerance mechanism developed for opportunistic grids environments [9]. This mechanism is based on the use of an autonomic checkpoint approach, which dynamically adjusts the time interval between successive checkpoints of a running task based on the node Mean Time Between Failures (MTBF) history, thus contributing to increase the success rate of the applications execution and, at the same time, reducing the cost involved in the checkpoint process.

Finally, we argue that the dynamism and instability of the resources that comprise opportunistic grids environments make them inappropriate for running applications with severe running time restrictions (hard deadlines) and, therefore, our approach is geared to meet soft applications execution time constraints (soft deadlines).

III. The Proposed Mechanism

The development of the approach presented in this paper was done in the context of the project InteGrade [1] [10], a multi-institutional effort to develop a middleware for opportunistic grids. In this middleware, the user informs constraints and preferences to be taken in consideration when submitting an application for execution. Constraints define minimum requirements for the selection of nodes, such as a specific hardware and software platform. Preferences are used to define an order in the selection of resources for the application execution, such as running it on machines with preferably more than 1 GB of main memory available. In this work, we also consider the users preferences with respect to time constraints for the application execution. The user is then able to specify whether the application being submitted belongs to one of the classes *nice* or *soft-deadline*. In the latter case, the user must provide a deadline for executing the desired application. For soft-deadline applications, the goal is the completion of applications within the informed extend of time. For the nice applications, the goal is just the application successful completion.

The application execution management mechanism for opportunistic grids proposed in this paper consists of the following components:

1) A prediction mechanism for the time execution of applications on the grid nodes;
2) An on-line scheduling heuristic which maps the application tasks to nodes that could potentially meet the application deadline as specified by it's user;
3) A mechanism that tracks the tasks execution progress on each grid node, checking whether or not is necessary to move them to other nodes in order to meet the specified deadline;
4) An adaptive application fault tolerance mechanism based on checkpoint, whose goal is to ensure the successful execution of applications, even in the event of failures.

Two of these four components were based on previous work, they are: the execution time prediction mechanism and the adaptive fault tolerance mechanism. The adopted execution time prediction mechanism is based on [11]. In this paper, the author presents two approaches for predicting the execution time of applications. In the first approach, the calculation of the estimated runtime is based on records of the application previous runs. The second approach, used in our work, is based on knowledge concerning the application execution model. The application code is analyzed, estimating the execution time of each task according to the capacity of the grid resources. Sun and Wu [12] developed a mathematical model to predict performance for a non-dedicated distributed environment. Their work was based on Gong et al. [13]. The prediction model took into consideration that the workstations comprising the distributed environment may be privately owned, which is exactly the case of opportunistic grid computing. In this way, parallel tasks submitted to the grid compete for execution with local sequential jobs submitted by the machines owners. The model also considers systems with heterogeneous machine utilization and heterogeneous service distribution and separates the influence of machine utilization, sequential job service rate, and parallel task allocation on the parallel completion time. A tacit assumption of the proposed model is that the parallel task can be partitioned freely into small pieces and it does not considered the effects of synchronization, communication, process migration, or granularity of parallelism. The adaptive application fault tolerance mechanism of is based on [9]. In this work, the authors present an autonomic strategy for application execution fault tolerance for opportunistic grids. The mechanism is based on two levels of adaptations: (a) parametric reconfiguration of fault tolerance strategies (checkpointing and replication); and (b) structural changes of the fault tolerance mechanism, by fully replacing the used technique. In our work, we explored the autonomic reconfiguration of the checkpoint technique, which dynamically adjusts the time interval between successive checkpoints of a running task based on the node MTBF history.

A component running on each grid node called Local Resource Manager (LRM) is responsible for tracking the execution progress of a task scheduled for its grid resource. This component receives from the grid scheduler the task execution request along with its respective constraints and properties, including its class (nice or soft-deadline) and the given deadline for its completion. Running tasks should regularly report the LRM about their execution progress using a well-defined API. On each received notification, the LRM estimates if the completion of the application should occur within the time limit and, if not, notifies the application through a callback method, forcing its suspension and the save of its state in a stable storage (checkpoint). The task execution request is then forwarded to the scheduler, which will map it to another grid node that could meet the requested deadline, if available.

The scheduling heuristic proposed in this paper follows an on-line approach, allowing the mapping of more than one task to a given grid node. The mapping of soft-deadline tasks is performed taking into account the nodes Mean Time Between Failure (MTBF) and their processing capacity. Soft-deadline tasks are scheduled to nodes that have been identified as stable (high MTBF) and whose capacity allows to conclude a task within the specified time constraint, as informed by the user during the application submission. Due to the cost imposed by the use of a checkpoint approach

[1] Homepage: http://www.integrade.org.br

and the possible local workload, a node is considered able for accomplishing the task if its capacity allows the task execution within the estimated deadline increased with a margin of 10%. If the local workload exceeds this initial estimative, the task execution monitoring mechanism (as described in the previous paragraph) is responsible for identifying the node inability in meeting the deadline and for re-submitting the task for a new scheduling.

The proposed approach includes a mechanism for advanced resource reservation that works as follows: when performing a task mapping, if there are no nodes available that meet the above criteria, the algorithm seeks for busy nodes that could satisfy the provided deadline and that the already running tasks meet the following conditions: (a) if the task class is nice, it will be suspended to make way for the soft-deadline task, reserving the resource for the later execution of the suspended nice task, that will be performed based on the last saved checkpoint; (b) if the task class is soft-deadline, the algorithm checks if the predicted remaining time for its execution increased with the time necessary to execute the task being scheduled is sufficient for accomplishing the provided deadline of the latter. If this condition holds, the resource is reserved for the execution of the task at hand, that will be carry out after the execution of the task already in place. Finally, if there are no resources that meet the above criteria, the user will have the application submission refused.

Nice tasks are scheduled for nodes considered less stable (with lower MTBF). This is done by ordering the available nodes according to a decreasing MTBF order and selecting the last one. If there are no nodes available, the algorithm searches for nodes running nice tasks and reserves the first node found for the later execution of the task being scheduled. Finally, if all the grid nodes are running soft-deadline tasks, the algorithm randomly chooses one, reserving it for executing the task after the already running computation has finish its execution.

IV. Evaluation

We evaluated our proposal through simulations that took into consideration various typical opportunistic grids scenarios, using as the evaluation metric the amount of soft-deadline applications executed within the user informed execution time restrictions. Since this work was done in the context of project InteGrade, we compared the results obtained with our approach with the regular InteGrade scheduling algorithm, that works as follows. The InteGrade scheduling algorithm [14] follows an on-line approach. It uses a filter to select resources based on constraints and preferences provided by users during the process of submitting applications. Constraints define minimum requirements for the selection of machines, such as hardware and software platforms, resource requirements such as minimum memory requirements. Preferences define the order used for choosing the resources, like rather executing on a faster CPU than on a slower one. The tasks that make up an application are then mapped to the nodes according to the ordered list of resources. If requirements and preferences are not specified, the algorithm maps the tasks to random chosen grid resources. The algorithm can map more than one task per node.

For performing the simulations, we used the AGST (Autonomic Grid Simulation Tool) [2] [15], an object-oriented discrete event simulator. The simulator provides, among others, tools for modeling grid resources and their network interconnections, grid applications and their submissions, the occurrence of resource faults, resources local workload, the use of workload and fault traces following the SWF (Standard Workload Format) [3] and FTA (Failure Trace Archive) [4] standards, a database model for storing relevant simulation generated data. Nevertheless, AGST major contribution is the definition and implementation of a simulation model based on the MAPE-K [16] autonomic management cycle, that can be used to simulate the monitoring, analysis and planning, control and execution functions, allowing the simulation of an autonomic computing grid. This is an important feature, since in our work we adopted an autonomic fault-tolerance mechanism.

A. Performed Experiments

The simulated grid environment comprises 100 machines within the same administrative domain, interconnected through a 100 Mbps network. In this environment, the average processing power is equivalent to a Pentium IV machine with 2.4 GHz (2,770 MIPS, considering the TSCP benchmark [5]), since we considered this as a good representative for personal computers. In order to take into consideration the environment resource heterogeneity, grid nodes were synthetically generated through an uniform distribution. We performed several simulations considering two heterogeneity factors: $U(1.385; 4.155)$ MIPS and $U(791; 4.746)$ MIPS. Using the first factor, the the processing power of the fastest machine is about 3 times greater then the processing power of the slowest one. In the later case, the difference is approximately 6 times.

The simulations took into consideration the existence of local workload in the grid resources. The defined workload model was based on Conde [17]. In this work, data regarding the use of resources (CPU and memory) from several machines belonging to laboratories of the Computer Science Department at the University of São Paulo were collected and stored in a trace file. By reading and analyzing these files, it was possible to simulate the workloads for both weekdays and for weekends. We developed an application

[2] http://www.lsd.ufma.br/∼agst

[3] http://www.cs.huji.ac.il/labs/parallel/workload/swf.html

[4] http://fta.inria.fr/

[5] http://home.comcast.net/∼tckerrigan/bench.html

that generates workload vectors, each one having 24 positions representing the 24 hours of a day. The vectors were passed as parameters to the AGST, that simulates the nodes workload.

For the grid workload, we synthetically generated a total of 770 regular grid applications for each experiment, varying their size (in millions of instructions) through an uniform distribution $U(39.888 \times 10^3; 159.552 \times 10^3)$ MI. Considering the average processing power of the grid machines (2,770 MIPS), each application execution would take 4 to 16 hours. We performed simulations taking into consideration several applications arrival rate per second: 0.002 (0.12 applications per minute); 0.004 (0.24 applications per minute) and 0.006 (0.36 applications per minute). During the simulations, we also varied the amount of soft-deadline applications comprising the simulated application set, using the following parameters: 25%, 50%, 75% and 100%. Table I summarizes the parameters used in the performed simulations.

Table I
SUMMARY OF THE SIMULATION PARAMETERS

machines (nodes)	100
heterogeneity factor	factor 3 = $U(1.385; 4.155)$ and factor 6 = $U(791; 4.746)$
regular grid applications	770 (tasks)
applications arrival rate	0.12; 0.24 and 0.36 (app/min)
percentage of soft applications	25; 50; 75 and 100 (%)
applications size in MI	4 to 16 hours = $U(39.888 \times 10^3; 159.552 \times 10^3)$

We performed a total of 48 simulations, by combining the two simulated scheduling strategies (our approach and the regular InteGrade one), the three applications arrival rate, the four amount of soft-deadline applications and the two environment resource heterogeneity factors. For each simulation, we generated 20 sets of 770 applications, leading to a total of 960 experiments.

Figure 1 presents the results obtained with the two scheduling strategies when using a 0.002 (0.12 applications per minute) arrival rate and an environment resource heterogeneity factor of the fastest machine being about 3 times greater than the processing power of the slowest one. As one can see, our proposed approach meets the runtime execution constraint of almost 100% of the soft-deadline submitted applications, even when 100% of the submitted application is of that class. This is approximately 25% better than the regular InteGrade algorithm, which accomplished almost 80% of the applications deadlines. In this case, both approaches presented a good performance, since the grid workload is relatively low.

Figure 2 shows the result in a scenario where the grid workload is higher, using an application arrival rate of 0.006 (0.36 applications per minute), maintaining the environment resource heterogeneity factor of the fastest machine being about 3 times greater than the processing power of the slowest one. As can be seen, in this case the regular InteGrade approach presents a worse result than was seen in the previous simulation, leading only to a 20% of deadlines accomplished when 50% of the submitted applications were soft-deadline. Our approach presented a much better result, meeting almost 50% of the informed deadlines, which represents a gain of 150%.

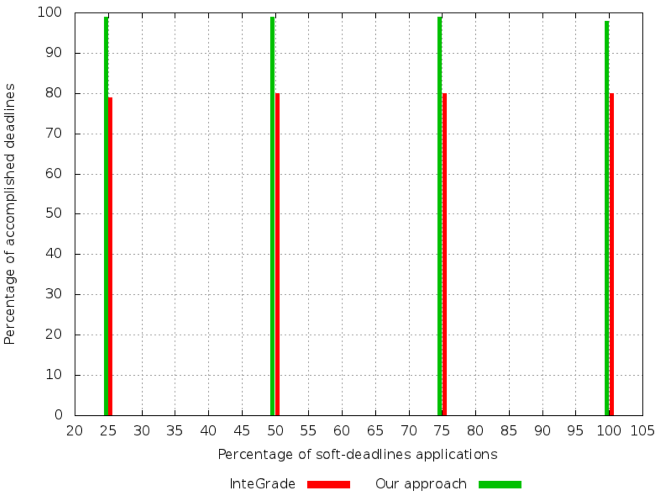

Figure 1. 0.12 applications per minute, heterogeneity factor 3

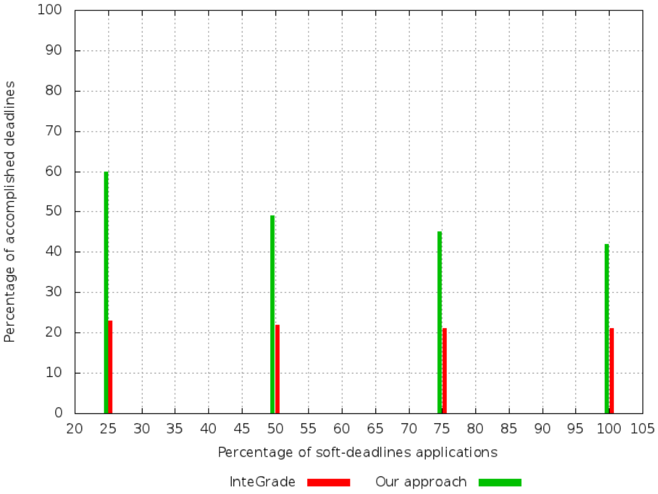

Figure 2. 0.36 applications per minute, heterogeneity factor 3

In another simulated scenario, we maintained the application arrival rate of 0.006 (0.36 applications per minute), altering the environment resource heterogeneity factor by having the fastest machine processing power being about 6 times greater than the slowest one. In this case, for a total of 50% of soft-deadline applications on each application set, our approach accomplished 60% of the requested deadlines, while the regular InteGrade achieved only 25%. This indicates that our approach can take advantage of the greater resource environment heterogeneity for achieving a higher performance.

Space constraints prevent us from showing the results of all the 48 performed simulations, but from the previous described ones, we can conclude that our approach performs much better than the regular InteGrade, considering the objective of running applications with runtime restrictions on opportunistic grid environments.

V. Conclusions and Future Work

Opportunistic grids execution environments are typically dynamic, heterogeneous, unpredictable and highly prone of failures. In addition, the resources of an opportunistic grid can be used at any time for the execution of local tasks, making it difficult to preview the conclusion of the tasks running on the grid nodes. These characteristics hinder the successful execution of applications for which there are time restrictions related to its completion.

This paper presented a new approach to application execution management considering user defined run time restrictions (soft deadlines) developed specifically for opportunistic grid environments. The proposed approach consists of a mechanism for predicting the execution time of applications in grid nodes, an on-line scheduling heuristic, a mechanism that tracks the execution progress of tasks running on the grid nodes and an adaptive fault tolerance mechanism based on the use of checkpoint. When properly combined, these mechanisms comprise a management model that allows applications to run within their time restrictions whenever possible, even in an environment as dynamic as the one typically provided by opportunistic grids. The proposed approach has been properly evaluated in a simulated environment, with experimental results demonstrating significant improvements when compared to traditional scheduling approaches used on computer grids.

In the future, we intend to explore and evaluate our proposal considering other classes of applications commonly used on computer grids, such as Bag-of-Tasks, Workflows and Parameter Sweep.

Acknowledgments

This work is supported by the Brazilian Federal Research Agency, CNPq, grant No.478853/2009-2, as well as by the state of Maranhão Research Agency (FAPEMA).

References

[1] D. da Silva, W. Cirne, and F. Brasileiro, "Trading cycles for information: Using replication to schedule bag-of-tasks applications on computational grids," in *Euro-Par 2003 Parallel Processing*, ser. Lecture Notes in Computer Science, H. Kosch, L. Böszörményi, and H. Hellwagner, Eds. Springer Berlin / Heidelberg, 2003, vol. 2790, pp. 169–180. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-45209-6_26

[2] H. Casanova, D. Zagorodnov, F. Berman, and A. Legrand, "Heuristics for scheduling parameter sweep applications in grid environments," in *Proceedings of the 9th Heterogeneous Computing Workshop*, ser. HCW '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 349–364. [Online]. Available: http://portal.acm.org/citation.cfm?id=795691.797922

[3] L. de Assis, "Uma heurística de escalonamento adaptativa à disponibilidade da informação para aplicações bag-of-tasks data-intensive em grids computacionais," Master's thesis, Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil, Setembro 2009. [Online]. Available: http://docs.computacao.ufcg.edu.br/posgraduacao/dissertacoes/2009/Dissertacao_LeonardodeAssis.pdf

[4] E. Santos-Neto, W. Cirne, F. Brasileiro, and A. Lima, "Exploiting replication and data reuse to efficiently schedule data-intensive applications on grids," in *Job Scheduling Strategies for Parallel Processing*, ser. Lecture Notes in Computer Science, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds. Springer Berlin / Heidelberg, 2005, vol. 3277, pp. 54–103. [Online]. Available: http://dx.doi.org/10.1007/11407522_12

[5] R. Buyya, M. Murshed, D. Abramson, and S. Venugopal, "Scheduling parameter sweep applications on global grids: a deadline and budget constrained cost-time optimization algorithm," *Software: Practice and Experience*, vol. 35, no. 5, pp. 491–512, April 2005. [Online]. Available: http://dx.doi.org/10.1002/spe.646

[6] J. Yu and R. Buyya, "Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms," *Sci. Program.*, vol. 14, no. 3,4, pp. 217–230, December 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1376960.1376967

[7] K. H. Kim, R. Buyya, and J. Kim, "Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters," in *Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, ser. CCGRID '07. Washington, DC, USA: IEEE Computer Society, May 2007, pp. 541–548. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2007.85

[8] S. S. Sathya and K. S. Babu, "Survey of fault tolerant techniques for grid," *Computer Science Review*, vol. 4, no. 2, pp. 101–120, May 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574013710000134

[9] A. E. B. Viana, B. de Tácio P. Gomes, J. F. Gonçalves, L. R. Coutinho, and F. J. da Silva e Silva, "Design and evaluation of autonomic fault tolerance strategies using the agst autonomic grid simulator," in *LatinAmerican Conference on High Performance and Distributed Computing (CLCAR '11)*, Colina, Mexico, Sep 2011.

[10] F. J. da Silva e Silva, F. Kon, A. Goldman, M. Finger, R. Y. de Camargo, F. C. Filho, and F. M. Costa, "Application execution management on the integrade opportunistic grid middleware," *Journal of Parallel and Distributed Computing*, vol. 70, no. 5, pp. 573 – 583, May 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731510000171

[11] Y. Liu, “Survey on grid scheduling,” Department of Computer Science, University of Iowa, for PhD Qualifying Exam, April 2004.

[12] X.-H. Sun and M. Wu, “Grid harvest service: A system for long-term, application-level task scheduling,” *Parallel and Distributed Processing Symposium, International*, vol. 0, p. 25a, april 2003. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/IPDPS.2003.1213102

[13] L. Gong, X.-H. Sun, and E. F. Watson, “Performance modeling and prediction of nondedicated network computing,” *IEEE Transactions on Computers*, vol. 51, no. 9, pp. 1041–1055, September 2002. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/TC.2002.1032624

[14] A. Goldchleger, F. Kon, A. Goldman, M. Finger, and G. C. Bezerra, “Integrade: Object-oriented grid middleware leveraging idle computing power of desktop machines,” *Concurrency and Computation: Practice and Experience*, vol. 16, no. 5, pp. 449–459, March 2004. [Online]. Available: http://dx.doi.org/10.1002/cpe.824

[15] B. de Tácio Pereira Gomes and F. J. da Silva e Silva, “Agst - autonomic grid simulation tool - a simulator of autonomic functions based on the mape-k model.” in *SIMULTECH*. SciTePress, 2011, pp. 354–359. [Online]. Available: http://dblp.uni-trier.de/db/conf/simultech/simultech2011.html#GomesS11

[16] M. C. Huebscher and J. A. McCann, “A survey of autonomic computing-degrees, models, and applications,” *ACM Comput. Surv.*, vol. 40, no. 3, pp. 7:1–7:28, August 2008. [Online]. Available: http://doi.acm.org/10.1145/1380584.1380585

[17] D. Conde, “Análise de padrões de uso em grades computacionais,” Master’s thesis, Departamento de Ciência da Computação - Universidade de São Paulo, Brasil, SP, Janeiro 2008, retrieved: 28/11/2011. [Online]. Available: http://www.integrade.org.br/files/danconde_dissertacao.pdf

Evaluation of the Severity of DoS Attacks on Computer Networks

Amar Aissani
Department of Computer Sciences
USTHB
Algiers, Algeria
aaissani@usthb.dz

Maroua Yaiche Achour
Department of Computer Sciences
USTHB
Algiers, Algeria
yaichemaroua@gmail.com

***Abstract*—In this paper, we derive stochastic recursive equations describing the evolution of a computer network under Denial-Of-Service (DoS) attacks (flooding attacks). The queue has several input processes, (i) the regular one (control packet flow and background or non control applications), which describes the input under network normal status; and (ii) the attack packet flow. We concentrate on some particular security measures, namely, the load or the loss probability. The load is strongly connected with the stability, which is understood as the convergence of the underlying stochastic process to a unique stationary ergodic regime. Loss of packets can occur although the system is stable. There are no specific requirements regarding statistical assumptions for establishing such equations. However, in order to derive stability condition and stationary performance measures we will need assumptions like stationarity and ergodicity (the independence is not required). Finally, we provide some numerical illustrations showing the effect of parameters on security measures and thus the severity of such attacks.**

***Keywords- Security; Queueing; Reliability; DoS Attack*.**

I. INTRODUCTION

Most of information systems are exposed to Denial-Of-Service (DoS) attacks, which is a one of the various forms of security threat of computer networks [1, 3, 9, 11]. The aim of DoS attacks is to exhaust a resource in the target system, that can be anything related to network computing and service performance (link bandwith, TCP connection buffers, application/service buffer, CPU times). Such attacks can also exploit a specific vulnerability in order to reduce or completely subvert the availability of the service provided. A common strategy used by an intruder to cause a DoS attack on a given target is to flood it with a continuous stream of packets that exhausts its connectivity. DoS attacks that use this kind of strategy are called brute-force attacks. Distributed Denial-of-service (DDoS) attacks are simply DoS attacks performed by multiple agents simultaneously. Many efforts have been made, in parallel with the evolution of DoS attacks, in the field of prevention and detection in networking security. Several countermeasures have been proposed, and can be roughly categorized as host-based systems and network based systems. Host-based systems are deployed on end-hosts and typically use firewall, intrusion detection systems (IDS) and/or balance the load among servers. This technique can help to protect the server, but not the legitimate access to the server because high-volume traffic may congest the incoming link to the server. Network-based systems are deployed inside networks (on routers) and fall into two categories: (1) Detection/identification mechanisms using signal processing and or statistical techniques; (2) Defense techniques using traffic control mechanisms such as egress or ingress filtering, route-based packet filtering disabling unusued services, and honeypots (see [1, 3, 9, 11]).

The Internet traffic is a complex stochastic process and there are several studies trying to describe a mathematical framework to model the behavior of these kinds of attacks. The interest of such models is to investigate how the attacks and other security anomalies affect the performance of the Network. It is difficult for legitimate users to launch real DoS attacks against the prototype of network to measure performance, since the attacks are themselves classified as cybercrime against the law.

We consider the model of NBCS (Network-Based Control System) described in [9] in which packets moving from one site to another have to access shared resources (communication links and network equipment). For each router in the path between a plant and a controller, the mechanism governing packet transmission can be abstracted by a queue with FIFO (First-in-First-Out) discipline of service. Packets arrive randomly at a router and can be modeled by some stochastic processes. If a packet finds the router CPU idle, it will be immediately served for a random amount of time. If the router CPU is busy, the packet will be in the queue to wait. When a queue with a finite size is full, the newly arrived packet enters “orbit” (a sort of queue) and repeat his attempt until he finds a place in the queue. Note that in the original version of [9] it is assumed that such a packet is dropped.

The routers in the path handle not only the NBCS packets flow, but also other traffic (non-control applications and flows of other NBCS systems). So, the model assumes several input processes: the first one is the regular (or

ISBN: 978-1-61208-198-4

legitimate) NBCS packet flow, the second one is the regular background traffic and the other is the DoS attack flow.

So, we consider that the server (i.e., the router CPU) has several inputs. The regular input describes the packet flow under normal network status. We separate the basic flow of control applications, which is assumed to be a Poisson process with rate λ_1 packets/sec. and the background flow (non control application/other packet flows), a Poisson process with rate ψ packets/sec. The attack traffic is modeled by a Poisson process with rate ϕ packets/sec. The Poisson assumption is conforming to some experimental studies in traffic studies [10] and also to some statistics about Denial of Service activities [11].

We denote by $\alpha = \{\sigma_n^1\}$, $\beta = \{\sigma_n^2\}$, $\delta = \{\sigma_n^3\}$ the sequences of service times for the control flow, the background flow and the attack flow respectively. We assume that these sequences form stationary ergodic sequences (or which is equivalent metrically transitive) without the usual independence assumption. The stationarity is understood here in the strict sense. The inter-retrial times are independent identically distributed random variables with common exponential distribution function with parameter $\nu > 0$.

In order to take into account the implemented defense mechanisms (firewall for example), we introduce a filtering parameter p, $0 < p < 1$. So, when a regular packet finds the service blocked, it is dropped with probability p. With probability $1-p$ it joins the service area (if it is not full) or a retrial group (also called an orbit). From the orbit it repeats his attempts at rate $\nu > 0$ until it gets service.

When a packet of the attack traffic finds the service blocked, it is dropped with probability q. It joins service area or orbit with probability $1-q$. In orbit, the attack packet evolves as a regular one: it adjusts its strategy and merges into regular packets. So, q can be seen as the filtering probability of an attack packet. For coherence, one can assume that $1 > q > p > 0$.

The paper is organized as follows. In Section II, we model the behavior of the network with a stochastic recursive sequence (SRS) (a generalization of the embedded Markov chain).

This representation gives an algorithm (Section III) for the simulation of sample paths of the underlying SRS and the statistical estimation of several security measures. We derive also the stability condition, which insures the existence of a unique stationary regime. There are no specific requirements regarding statistical assumptions for establishing such equations. However, in order to derive stability condition and stationary performance measures we will need assumptions like stationarity and ergodicity (the independence is not required).

In Section IV, we adjust the model by considering some other types of attacks which conduct to the interruptions of service.

Finally, we provide (Section V) some numerical illustrations showing the effect of parameters on security measures and thus the severity of such attacks.

II. A STOCHASTIC EQUATION FOR SIMULATING NETWORK SAMPLE PATHS

In a first part, we assume that the buffer $K = 0$. Let $\{N(t), t \geq 0\}$ be the number of packets in the orbiting queue at time t. It represents a stochastic process on the discrete space of natural integers. Let $C(t)$ be another 3-valued random process describing the server status: $C(t) = 0$ if the server is free at time t; $C(t) = i$ if the server is busy by service of a certain packet of type i at time t, $i = 1,2,3$.

We consider the process $\{N_n\}$ embedded immediately after service times γ_n (i.e., $N_n = N(\gamma_n + 0)$). Denote by $X_n = (C_n, N_n), n \geq 1$ the sequence of successive states of the system at these epochs where $C_n = C(\gamma_n + 0)$. Observe that if the sequence α, β, δ are independent and identically distributed, then the sequence $\{X_n, n \geq 1\}$ forms a Markov chain (in the usual sense) defined on the state space $S = \{1,2\} \otimes IN$ and the ergodicity condition can be derived using the Foster-Moustafa-Tweedie criterion [5].

We next show that the process $\{N_n\}$ is a stochastic Recursive Sequence in the sense of Borovkov [4, 5]. Recall that a process $\{N_n\}$ is called a SRS with driver $\{\{\xi_n\}, f\}$, if for some function f it satisfies the equation $N_{n+1} = f(N_n, \xi_n), \forall n \geq 0$ where the driving sequence ξ_n is a stationary ergodic stochastic process.

It is well known that for the classical FIFO queue, idle (respectively, busy) server period coincides with the system idle (respectively, busy) period. It is not the case for retrial queues where in the system busy period the systems evolves as an alternating sequence of idle periods and busy periods of the server.

The situation is slightly different in the case of finite buffer. Let τ_n be the nth idle server period, i.e., the time between the end of the $n-1$th service till the beginning of the nth service. The distribution of τ_n is determined by the competition between inter arrival times and inter retrial times, which event occurs first. This idle period ends when either there is an external arrival (regular or attacker) or when a call from orbit tries to retry. Under our assumptions, τ_n is exponentially distributed with parameter $\lambda+\nu$ where $\lambda=(\lambda_1+\psi)(1-p)+\phi(1-q)$ (constant retrial policy) and $\lambda+\nu N_n$ (linear retrial policy). The conditional probability, given $\Im(\gamma_n)$ (the sigma algebra generated by events describing state of the system up to time γ_n), that the first event to occur after the $n-1$th service ends (and after the served packet has left the system) is an external arrival (regular or attacker), equals $\frac{\lambda}{\lambda+\nu}$ (respectively, $\frac{\lambda}{\lambda+\nu N_n}$). The conditional probability, given $\Im(\gamma_n)$ that the first event to occur after the $n-1$th service ends (and after the served packet has left the system) is a retrial, equals $\frac{\nu}{\lambda+\nu N_n}$ (respectively, $\frac{\nu N_n}{\lambda+\nu N_n}$).

Consider the following two Pseudo Random Generators, $u_n^1=\{u_n^1, n=0,1,...\}$ and $u_n^2=\{u_n^2, n=0,1,...\}$ They are described in fact by two sequences of random variables distributed uniformly on $[0,1]$ mutually independent, and independent of the sequences α,β,δ. Let $\chi(A)$ be the characteristic function of the event A : $\chi(A)=1$, if A has occurred, $\chi(A)=0$ otherwise. We will need also a mean to generate the input Poisson random processes.

For the formal description below we introduce an application $\Pi : IR^+\times[0,1]\to IN$ defined by

$$\Pi(t,x)=\inf\left\{n\in IN : \sum_{k=0}^{n}\frac{t^k e^{-t}}{k!}\geq x\right\}.$$

Thus, $\Pi(t,u_n^1)$ implements a Random Poisson Generator for the sequences the instant of primary arrival packets (regular or attacker) or secondary (retrials) [2, 12]. The second sequence $u_n^2=\{u_n^2, n=0,1,...\}$ will be used to generate which event has occurred. Formally, we can consider the following events G_n, H_n, S_n, R_n such that

$$G_n=\left\{0\leq u_n^2\leq\frac{\lambda_1(1-p)}{\lambda+\nu N_n}\right\},$$

$$H_n=\left\{\frac{\lambda_1(1-p)}{\lambda+\nu N_n}\leq u_n^2\leq\frac{(\lambda_1+\psi)(1-p)}{\lambda+\nu N_n}\right\},$$

$$S_n=\left\{\frac{(\lambda_1+\psi)(1-p)}{\lambda+\nu N_n}\leq u_n^2\leq\frac{(\lambda_1+\psi)(1-p)+\phi(1-q)}{\lambda+\nu N_n}\right\},$$

$$R_n=\left\{\frac{(\lambda_1+\psi)(1-p)+\phi(1-q)}{\lambda+\nu N_n}\leq u_n^2\leq 1\right\}.$$

According to the relations above, we have the following stochastic equation

$$N_{n+1}=\max(0,N_n+\xi_n)=(N_n+\xi_n)^+ \qquad (2.1)$$

where $\xi_n=h(N_n,\sigma_n^1,\sigma_n^2,\sigma_n^3,u_n^1,u_n^2)$ is given by

$$\xi_n=\chi(G_n)\Pi(\lambda\sigma_n^1,u_n^1)+\chi(H_n)\Pi(\lambda\sigma_n^2,u_n^1)+\chi(S_n)$$

$$\Pi(\lambda\sigma_n^3,u_n^1)+\chi(R_n)(\Pi(\lambda\sigma,u_n^1)-1) \qquad (2.2)$$

for linear retrials. In the case of constant retrial rate, the equations (1)-(2) remains valid except that the term νN_n is replaced by ν in the definition of the events G_n, H_n, S_n, R_n.

Formula (2.1) is just an arithmetical count of the number of customers in the system at a given time. The number of customers N_{n+1} in orbit after the n+1 service equal the number of customers N_n at the previous nth service time plus the variable ξ_n. This variable counts the difference between the number of arrivals and departures during the period $[\gamma_n,\gamma_{n+1}]$ (interval between the two successive departures nth and n+1th. The operator max stay here, since the variable ξ_n cannot be negative.

The first term in formula (2.2) counts the number of packets of the basic flow (control applications), which have been accepted by the filter (with probability 1-p), i.e., when the event G_n occurs; the second term counts the number of packets of the background flow, which have been accepted by the filter(also with probability 1-p) (when H_n occurs); the third term counts the number of packets of the attack flow, which have been accepted by the filter (with probability 1-q), i.e., when the event S_n occurs); finally, the forth term counts the number of packets which has been served and exit the systems (when the event R_n occurs).

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-198-4

In both cases, the process

$$\xi_n = \left(\sigma_n^1, \sigma_n^2, \sigma_n^3, u_k^1, u_k^2, k \le n\right)$$

is the driving sequence for the SRS taking values in $\Theta = IR^+ \otimes IR^+ \otimes IR^+ \otimes [0,1] \otimes [0,1]$ and is assumed stationary ergodic.

In (2.1)-(2.2) it is assumed that the buffer K=0, while the original model [9] assume a finite buffer of capacity $K \ge 1$. In this case, a retrial occurs if a packet finds the buffer full. So, the stochastic equation (1) needs to be refined.

Let M_n be the number of packets in the buffer at time γ_n, and then the basic process is now described by the two-dimensional process $Y_n = (M_n, N_n)$. In this case, the SRS has the following form

If $M_n < K, \xi_n \le K - M_n$, then

$$\left(M_{n+1}, N_{n+1}\right) = \left(M_n + \xi_n, N_n\right) \qquad (3.1)$$

If $M_n = K, \xi_n > K - M_n$, then

$$\left(M_{n+1}, N_{n+1}\right) = \left(K, N_n + \xi_n - (K - M_n)\right). \qquad (3.2)$$

The process Y_n describe the behavior of the network when $K \ge 1$, but finite. The above SRS shows how to compute Y_n. We distinguish two cases. Formula (3.1) corresponds to the case when the buffer is not full and formula (3.2) to the case when the buffer is full, i.e., $M_n = K$.

III. Simulation Algorithm and It's Perormance

The representation under the form of SRS is particularly adapted to a discrete-event simulation of the network under DoS attacks.

```
Set  $N_0 = 0$ (initialization)
Repeat
$u^1 \leftarrow$ Random; {generation of $u^1$ and arrival event}
$u^2 \leftarrow$ Random; {generation of $u^2$ and the type of the arrival packet}
$u^3 \leftarrow$ Random; {generation of $u^3$ and the service time random variable according to the given probability distribution}
Poisson variables are generated using any algorithm for Poisson process.
For all n,
Computation of $\xi_n = f(u^1, u^2, \sigma^i), i = 1,2,3$ by formula (2.2) or (2.3).
Computation of $N_{n+1} = \max(0, N_n + \xi_n) = (N_n + \xi_n)^+$.
Computation of the state at time $n+1$ given $N_n$ and $\xi_n$.
End for
Until  $n < T$  ($T =$ end of simulation).
```

Based on the SRS formulation,, the above algorithm gives directly a sample of the steady state distribution provided the network is stable (see Fig. 1 in Section V). Next, we can compute statistical estimate of any security metric directly from sample paths (Delay, Loss probability, Load…

In fact, the algorithm simulate the physical operation of the system, arriving customers (regulars or attackers), retrial requests, filtering actions and service of customers. It handle these different actions by the next-event incrementing procedure, which differs from the fixed-time incrementing in that the master clock is incremented by a variable amount rather than by a fixed amount of time.

Conceptually, the next-event incrementing procedure is to keep the simulated system running without interruption until an event occurs, at which point the algorithm pauses momentarily to record the change in the system. To implement this idea, the algorithm actually proceeds by keeping track of when the next few simulated events are scheduled to occur, jumping in simulated time to the first of these events, and updating the system. The cycle ends at time T and it is repeated as many time as desired, say N times.

We can see that the running time is $N \times T$ unit of times.

It is important to estimate the quality of the estimation of mean performance measures. The precision is $\frac{1}{\sqrt{N}}$ by the law of large numbers [2, 12].

We can prove [3] that the network is stable if $\rho < 1$, ,where

$$\rho = \frac{\lambda + \nu}{\nu} \times$$
$$\times \left[\lambda_1^2 (1-p) E(\sigma^1) + \psi^2 (1-p) E(\sigma^2) + \phi^2 (1-q) E(\sigma^3)\right]$$

in the case of constant retrials and

ISBN: 978-1-61208-198-4

$$\rho = \lambda \times \left\lfloor \lambda_1^2 (1-p)E(\sigma^1) + \psi^2 (1-p)E(\sigma^2) + \phi^2 (1-q)E(\sigma^3) \right\rfloor$$

in the case of linear retrials.

Here, the stability is understood as the strong coupling convergence [3-6, 8]) to a unique stationary regime. This condition is also a condition of convergence of the algorithm of Section III. The formula for ρ depends on the retrial policy (constant or linear). This quantity represents the traffic intensity and also the load, which will serve here as a security measures for detecting the status of the network: normal or under attack. The network is under attack if the value of ρ crosses a given threshold.

Another security metric, which is not considered here is the Loss probability, when $K \geq 1$. In this case, we can detect a DoS attack if the loss probability (depending on K) is large. So, the security status is defined by a threshold $\varepsilon = \varepsilon(K) > 0$ small enough. The network is under DoS attack if the loss probability is $\geq 1 - \varepsilon$. An application of such security measure can be found in the work [1] with a different model.

IV. ATTACKS ON THE AVAILABILITY

We have up now considered DoS attacks, more precisely flooding attacks which aim to saturate the system by sending many requests of service. But, there is another type of attacks which exploit a specific vulnerability in order to reduce or completely subvert the availability of the service provided (interruption of service). In this section we take into account such attacks in the previous model by introducing a new parameter θ, the rate of such attacks. So, we assume that the service becomes unavailable for a random restoration period of time. Such attacks occur according to a Poisson process with rate θ. We denote by $r^{(n)} = \left\{ r_i^{(n)}, i = 1,2,... \right\}$ the sequence of "renewal" (restoration to the as-good-as new state) times, which is assumed again stationary ergodic and independent of the other sequences of parametric random variables. In this case, we have again the representation of the basic process under the form of SRS (2.1). We need only to take into account delay due to renewal times and the incrementation of DoS attacks during such periods :

$$\sum_{i=1}^{\Pi(\theta\sigma, u_n)} \Pi\left(\omega r_i^{(n)}, u_i^{(n)} \right), \tag{4.1}$$

where $\omega = \lambda_1 (1-p) + \phi(1-p)$ or $\psi(1-q)$ according to the case which occurs.

Formula (4.1) indicates that the full service of a given customer (if it is not lost) is the pure service plus the cumulated duration of all interruptions occurring during this service.

The model can also take into account other types of interruptions, for example due to software or hardware failures.

V. NUMERICAL ILLUSTRATIONS

In this section, we show the effect of DoS attack on some security measures. First, Fig. 1 shows some sample paths of the stochastic process $\{N_n\}$ and the simulation algorithm of Section III.

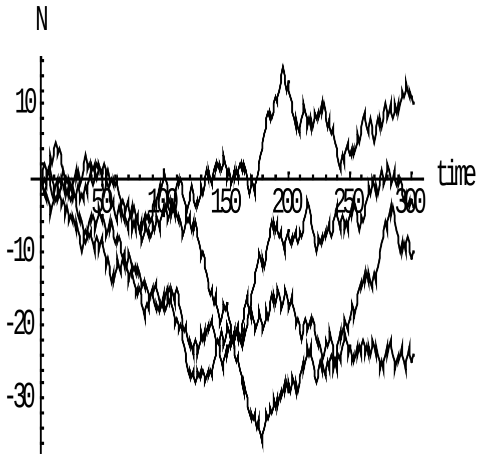

Figure. 1. Sample paths of the process $\{N_n\}$.

From these sample paths, we can compute the sample average of any security measure (for example, loss probability, etc.), which is an estimation of the true security measure. This estimation is unbiased, consistent and efficient (in the statistical sense) [2 , 12].

Fig. 2 compares the evolution of the load ρ as a function of the attack parameter ϕ. We neglect the background flow ($\psi = 0$) and fix some parameters. We assume the mean service times are identical for all types of requests and set $\lambda_1 = 10/\sec$. We observe that the load increases with the severity of the attack (when the attack rate increases) for a fixed value of the retrial rate. The load decreases with increasing of the retrial rate.

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-198-4

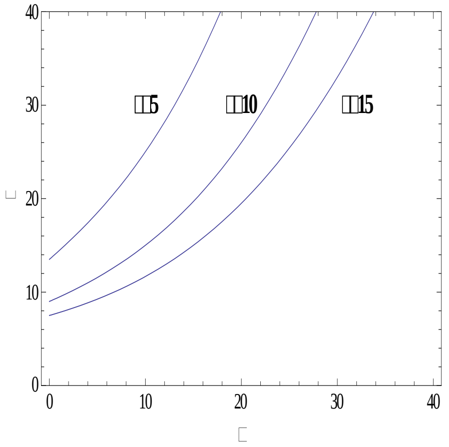

Figure 2. Effect of DoS attack rate ϕ on the load ρ for different values of the retrial rate ν.

Fig. 3 is another view of this observation. It shows the effect of the retrial rate ν on the load for different values of the attack parameter ϕ. We consider three cases $\phi = 0$ (under normal network status), $\phi = 20$ or $\phi = 40$ (under DoS attack).

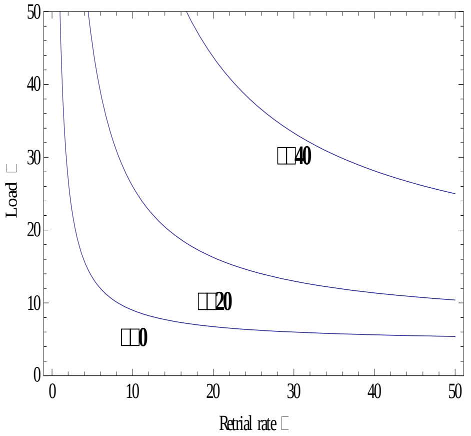

Figure 3. Effect of retrial rate ν on the Load ρ for Different Values of ϕ.

We observe that the load decreases with increasing of the retrial rate ν. The load increases with the severity of attack.

VI. CONCLUSION

In this paper, we have provided an extension of the model of [1], which takes into account the possibility of retrials of packets and also the existence of defense mechanisms (firewalls). The evolution of the system is described by a stochastic recursive sequence, which provides a practical mean to simulate sample paths of the underlying process and estimate several security measures. Such a security measure serves as an indicator of intrusion. The stability condition is obtained under quite general assumptions about service process (stationarity in the strict sense and ergoditicity). The model can be refined by taking into account some other phenomena and also the comparison with real data. Although it is a practice to assume Poisson arrivals in a first study, it will be interesting to consider the case of non Poisson arrivals as reported in some experimental studies.

ACKNOWLEDGMENT

We are very grateful to the reviewers and the editor for their helpful comments and suggestions. This research was supported by the Ministry of High Education and Scientific Research.

REFERENCES

[1] A. Aissani, "Queueing Analysis for Network Under DoS Attacks," In Lecture Notes in Theoretical Computer Science, O. Gervasi and al., Ed. Springer Heidelberg, Berlin, vol. 5073, Part II, pp. 500-513, 2008.

[2] A. Aissani, Modeling and Simulation. 2nd Edition, Office of University Publications (OPU), Algiers, 2010, (in French).

[3] A. Aissani, "Stochastic Analysis of a Network under DoS Attacks," unpublished.

[4] E. Altman, "On the Stability of Retrial Queues," Queueing Sys. vol. 26, no 3-4, pp. 343-363, 1997.

[5] A. Borovkov, Ergodicity and Stability of Stochastic Processes, Wiley, New York, 1998.

[6] A. Borovkov and S.G. Foss, "Stochastic Recursive Sequences and their Generalizations," Siberian Advances in Mathematics. vol. 2, pp. 16-81, 1992.

[7] G. Falin and J.G.C. Templeton, Retrial Queues, Chapman and Hill, New Jersey, 1997.

[8] T. Kernane and A. Aissani, "Stability of Retrial Queues with Versatile Policy," Appl. Math. & Stoch. Analysis. Article ID 54359, pp. 1-16, 2006.

[9] M. Long, J. Chawan-Hwa and J. Hung, "Denial of Service Attacks on Network-Based Control Systems: Impact and Mitigation," IEEE. Transactions on Industrial Informatics, vol. 1, no 2, pp. 85-96, 2005.

[10] R. Martin , Basic Traffic Analysis, Prentice-Hall, New Jersey, 1993.

[11] D. Volker, G., Savage, "Inferring Internet Denial-of-Service Activity," Proc. UNESIX Security Symposium, pp. 9-22, IEEE Press, New York, 2001.

[12] P. Tavel, Modeling and Simulation Design. AK Peters Ltd., Natick, MA , 2007.

Robustness of Random Graphs Based on Natural Connectivity

Jun Wu, Yuejin Tan, Hongzhong Deng
College of Information Systems and Management
National University of Defense Technology
Changsha 410073, P. R. China
Email: wujunpla@hotmail.com, yjtan@nudt.edu.cn, hongzhongdeng@163.net

Mauricio Barahona
Department of Mathmatics
Imperial College London
London SW7 2AZ, United Kingdom
Email: m.barahona@imperial.ac.uk

***Abstract*—Recently, it has been proposed that the natural connectivity can be used to efficiently characterize the robustness of complex networks. Natural connectivity quantifies the redundancy of alternative routes in a network by evaluating the weighted number of closed walks of all lengths and can be regarded as the average eigenvalue obtained from the graph spectrum. In this paper, we explore the natural connectivity of random graphs both analytically and numerically and show that it increases linearly with the average degree. By comparing with regular ring lattices and random regular graphs, we show that random graphs are more robust than random regular graphs; however, the relationship between random graphs and regular ring lattices depends on the average degree and graph size. We derive the critical graph size as a function of the average degree, which can be predicted by our analytical results. When the graph size is less than the critical value, random graphs are more robust than regular ring lattices, whereas regular ring lattices are more robust than random graphs when the graph size is greater than the critical value.**

***Keywords*-natural connectivity; robustness; complex networks; random graphs; regular graphs.**

I. Introduction

Networks are everywhere. Many systems in nature and society can be described as complex networks. Examples of networks include the Internet [1], metabolic networks [2], electric power grids [3], supply chains [4], urban road networks [5], world trade web [6] and many others. Complex networks, more generally, complex systems have become pervasive in today's science and technology scenario and have recently become one of the most popular topics within the interdisciplinary area involving physics, mathematics, biology, social sciences, informatics, and other theoretical and applied sciences (see [7]–[9]). Complex networks rely for their function and performance on their robustness, that is, the ability to endure threats and survive accidental events. Due to their broad range of applications, the attack robustness of complex networks has received growing attention.

Recently, we showed that the concept of natural connectivity can be used to characterize the robustness of complex networks [30]. The concept of natural connectivity is based on the Estrada index of a graph, which has been proposed to characterize molecular structure [31], bipartivity [32], subgraph centrality [33] and expansibility [34], [35]. Natural connectivity has an intuitive physical meaning and a simple mathematical formulation. Physically, it characterizes the redundancy of alternative paths by quantifying the weighted number of closed walks of all lengths leading to a measure that works in both connected and disconnected graphs. Mathematically, the natural connectivity is obtained from the graph spectrum as an average eigenvalue and it increases strictly monotonically with the addition of edges. Abundant information about the topology and dynamical processes can be extracted from a spectral analysis of the networks. Natural connectivity sets up a bridge between the graph spectra and the robustness of complex networks and receives growing attention [36]–[38]. In our previous study [39], we have shown that the natural connectivity of regular ring lattices is independent of the network size and increases linearly with the average degree. In this paper, we investigate the natural connectivity of random graphs and compare it with regular graphs.

The paper is structured as follows. In Section 2, we introduce the concept of natural connectivity and some basic elements of random graphs. In Section 3, we derive the natural connectivity of random graphs. In Section 4, we compare the natural connectivity of random graphs with that of regular graphs. Finally, the conclusions are presented in Section 5.

II. Related work

Simple and effective measures of robustness are essential for the study of robustness. A variety of measures, based on different heuristics, have been proposed to quantify the robustness of networks. For instance, the vertex (edge) connectivity of a graph is an important, and probably the earliest, measure of robustness of a network [10]. However, the vertex (edge) connectivity only partly reflects the ability of graphs to retain connectedness after vertex (or edge) deletion. Other improved measures include super connectivity [11], conditional connectivity [12], restricted connectivity [13], fault diameter [14], toughness [15], scattering number [16], tenacity [17], the expansion parameter [18] and the isoperimetric number [19]. In contrast to vertex (edge) connectivity, these new measures consider both the cost to damage a network and how badly the network is damaged. Unfortunately, from an algorithmic point of view, the problem of calculating these measures for general graphs is NP-complete. This implies that these measures are of no great practical use within the context of complex networks. Another remarkable measure used to unfold the robustness of a network is the second smallest (first non-zero) eigenvalue of the Laplacian matrix, also known as the algebraic connectivity. Fiedler [20] showed that the magnitude of the algebraic connectivity reflects how well connected the overall graph is; the larger the algebraic connectivity is, the more difficult it is to cut a graph into independent components. However, the algebraic connectivity is equal to zero for all disconnected networks. Therefore, it is too coarse a measure to be used for complex networks..

An alternative formulation of robustness within the context of complex networks emerged from the random graph theory [21] and was stimulated by the work of Albert *et al.* [22]. Instead of a strict extreme property, they proposed a statistical measure, that is, the critical removal fraction of vertices (edges) for the disintegration of a network, to characterize the robustness of complex networks. The disintegration of networks can be observed from the decrease of network performance. The most common performance measurements include the diameter, the size of the largest component, the average path length, the efficiency [23] and the number of reachable vertex pairs [24]. As the fraction of removed vertices (or edges) increases, the performance of the network will eventually collapse at a critical fraction. Although we can obtain the analytical critical removal fraction for some special networks [25]–[29], generally, this measure can only be calculated through simulations.

III. Preliminaries

A. Graph and Natural Connectivity

A complex network can be viewed as a simple undirected graph $G(V,E)$, where V is the set of vertices, and $E \subseteq V \times V$ is the set of edges. Let $N = |V|$ and $M = |E|$ be the number of vertices and the number of edges, respectively. Let $A(G) = (a_{ij})_{N\times N}$ be the adjacency matrix of G, where $a_{ij} = a_{ji} = 1$ if vertices v_i and v_j are adjacent, and $a_{ij} = a_{ji} = 0$ otherwise. It is obvious that $A(G)$ is a real symmetric matrix. We thus let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$ denote the eigenvalues of A which are usually also called the eigenvalues of the graph G itself. The set $\{\lambda_1, \lambda_2, ...\lambda_N\}$ is called the spectrum of G. The spectral density of G is defined as the sum of the δ functions as follows

$$\rho(\lambda) = \frac{1}{N}\sum_{i=1}^{N}\delta(\lambda - \lambda_i) \tag{1}$$

which converges to a continuous function when $N \to \infty$, where $\delta(\lambda - \lambda_i) = 1$ if $\lambda = \lambda_i$; and $\delta(\lambda - \lambda_i) = 0$, otherwise.

A walk of length k in a graph G is an alternating sequence of vertices and edges $v_0e_1v_1e_2...e_kv_k$, where $v_i \in V$ and $e_i = (v_{i-1}, v_i) \in E$. A walk is closed if $v_0 = v_k$. The number of closed walks is an important index for complex networks. Recently, we have defined the redundancy of alternative paths as the number of closed walks of all lengths [30]. Considering that shorter closed walks have more influence on the redundancy of alternative paths than longer closed walks and to avoid the number of closed walks of all lengths to diverge, we scale the contribution of closed walks to the redundancy of alternative paths by dividing them by the factorial of the length k. That is, we define a weighted sum of numbers of closed walks $S = \sum_{k=0}^{\infty} n_k/k!$, where n_k is the number of closed walks of length k. This scaling ensures that the weighted sum does not diverge and it also means that S can be obtained from the powers of the adjacency matrix:

$$n_k = trace(A^k) = \sum_{i=1}^{N}\lambda_i^k \tag{2}$$

Using Eq. 2, we can obtain

$$S = \sum_{k=0}^{\infty}\frac{n_k}{k!} = \sum_{k=0}^{\infty}\sum_{i=1}^{N}\frac{\lambda_i^k}{k!} = \sum_{i=1}^{N}\sum_{k=0}^{\infty}\frac{\lambda_i^k}{k!} = \sum_{i=1}^{N}e^{\lambda_i}. \tag{3}$$

Hence, the proposed weighted sum of closed walks of all lengths can be derived from the graph spectrum. We remark that Eq. 3 corresponds to the Estrada Index of the graph [31], which has been used in several contexts in the graph theory, including bipartivity [32] and subgraph centrality [33]. The natural connectivity can be defined as the average eigenvalue of the graph, as follows.

Definition [30] Let $A(G)$ be the adjacency matrix of G and let $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N$ be the eigenvalues of $A(G)$. Then the natural connectivity or natural eigenvalue of G is defined by

$$\bar{\lambda} = \ln\left(\sum_{i=1}^{N}e^{\lambda_i}/N\right) \tag{4}$$

It is evident from Eq. 4 that $\lambda_N \leq \bar{\lambda} \leq \lambda_1$.

A regular ring lattice $RRL_{N,2K}$ is a $2K-regular$ graph with N vertices on a one-dimensional lattice, in which each vertex is connected to its $2K$ neighbors (K on either side). In a previous study [39], we have investigated the natural connectivity of regular ring lattices and shown that random regular graphs are less robust than regular ring lattices based on natural connectivity.

Theorem III.1. *[39] Let $RRL_{N,2K}$ be a regular ring lattice.Then the natural connectivity of $RRL_{N,2K}$ is*

$$\bar{\lambda}_{R_{N,2K}} = \ln\left(I_0(\overbrace{2,2,...2}^{K})\right) + o(1) \tag{5}$$

where $o(1) \to 0$ as $N \to \infty$.

B. Erdős-Rényi Random Graphs

Random graphs have long been used for modelling the topology of systems made up of large assemblies of similar units. The theory of random graphs was introduced by Erdős *et al.* [40]. A detailed review of random graphs can be found in the classic book [21]. A random graph is obtained by starting with a set of N vertices and adding edges between them at random. In this paper, we study the random graphs of the classic Erdős-Rényi model $G_{N,p}$, in which each of the possible $C_N^2 = N(N-1)/2$ edges occurs independently with probability p. Consequently, the total number of edges M is a random variable with the expectation value $E(M) = p{\cdot}C_N^2$ and then the average degree $<k>=(N-1)p \approx Np$.

Random graph theory studies the properties of the probability space associated with graphs with N vertices as $N \to \infty$. Many properties of such random graphs can be determined using probabilistic arguments. We say that a graph property Q holds almost surely for $G_{N,p}$ if the probability that $G_{N,p}$ has property Q tends to one as $N \to \infty$. Furthermore, Erdős *et al.* described the behavior of $G_{N,p}$ very precisely for various values of p [41]. Their results showed that:

1) If $Np < 1$, then a graph $G_{N,p}$ will almost surely have no connected components of size larger than $o(\ln N)$; If $Np \geq 1$, then a graph $G_{N,p}$ will almost surely have a unique "giant" component containing a positive fraction of the vertices.
2) If $Np < \ln N$, then a graph $G_{N,p}$ will almost surely not be connected; If $Np \geq \ln N$, then a graph $G_{N,p}$ will almost surely be connected.

It is well known that the largest eigenvalue λ_1 of $G_{N,p}$ is almost surely $Np[1+o(1)]$ provided that $Np \gg \ln N$(see [42], [43]). Moreover, according to the famous Wigner's law or semicircle law [44], as $N \to \infty$, the spectral density of $G_{N,p}$ converges to a semicircular distribution as follows

$$\rho(\lambda) = \begin{cases} \frac{2\sqrt{R^2-\lambda^2}}{\pi R^2} & |\lambda| \leq R \\ 0 & |\lambda| > R \end{cases} \tag{6}$$

where $R = 2\sqrt{Np(1-p)}$ is the radius of the "bulk" part of the spectrum.

IV. Natural connectivity of random graphs

When $N \to \infty$, with continuous approximation for λ_i, Eq. 4 can be rewritten in the following spectral density form

$$\bar{\lambda} = \ln\left(\int_{-\infty}^{+\infty} \rho(\lambda) e^{\lambda} d\lambda\right) = \ln\left(M_\lambda(1)\right) \tag{7}$$

where $\rho(\lambda)$ is the spectral density and $M_\lambda(t)$ is the moment generating function of $\rho(\lambda)$. Consequently, we obtain the natural connectivity of Erdős-Rényi random graphs with $p \gg \ln N/N$

$$\bar{\lambda} = \ln\left(\int_{-R}^{+R} \rho(\lambda) e^{\lambda} d\lambda + e^{\lambda_1}/N\right) = \ln\left(M_\lambda(1) + e^{Np}/N\right) \tag{8}$$

where

$$M_\lambda(1) = \int_{-R}^{+R} \frac{2\sqrt{R^2-\lambda^2}}{\pi R^2} e^{\lambda} d\lambda = \frac{2}{\pi}\int_{-R}^{+R} \frac{\sqrt{R^2-\lambda^2}}{R^2} e^{\lambda} d\lambda \tag{9}$$

Substituting $\lambda = R\cos(\theta)$ into Eq. (8), we obtain that

$$M_\lambda(1) = \frac{2}{\pi}\int_0^{\pi} e^{R\cos(\theta)} \sin^2(\theta) d\theta \tag{10}$$

Note that [45]

$$I_\alpha(x) = \frac{(x/2)^\alpha}{\pi^{1/2}\Gamma(\alpha+1/2)}\int_0^{\pi} e^{x\cos(\theta)} \sin^{2\alpha}(\theta) d\theta \tag{11}$$

where $I_\alpha(x)$ is the modified Bessel function and $\Gamma(x)$ is the Gamma function. Then we obtain that

$$I_1(R) = \frac{R}{\pi}\int_0^{\pi} e^{R\cos(\theta)} \sin^2(\theta) d\theta \tag{12}$$

Using Eq. (11), we can simplify Eq. (9) as

$$M_\lambda(1) = 2I_1(R)/R \tag{13}$$

Substituting Eq. (12) into Eq. (7), we obtain that

$$\bar{\lambda} = \ln\left(\frac{2I_1(R)}{R} + \frac{e^{Np}}{N}\right) = Np - \ln(N) + \ln\left(1 + \frac{2NI_1(R)}{e^{Np}R}\right) \tag{14}$$

Now we propose two lemmas first.

Lemma IV.1. *As* $N \to \infty$, $f(p) = 2NI_1(R)/(e^{Np}R)$ *is a monotonically decreasing function for* $\ln N/N < p < 1 - \ln N/N$, *where* $R = 2\sqrt{Np(1-p)}$.

Proof: It is easy to know that $2\sqrt{\ln N(1-\ln N/N)} < R \leq \sqrt{N}$ for $\ln N/N < p < 1 - \ln N/N$. Then as $N \to \infty$, we have $R \to \infty$. Note that, for the large values of $x \gg |\alpha^2 - 1/4|$, the modified Bessel functions $I_\alpha(x)$ have the following asymptotic forms [46]

$$I_\alpha(x) \to \frac{1}{\sqrt{2\pi x}} e^x \tag{15}$$

Thus, for $\ln N/N < p < 1 - \ln N/N$, we obtain

$$I_1(R) \to \frac{1}{\sqrt{2\pi R}} e^R \tag{16}$$

Then, we have

$$f(p) \to N\sqrt{\frac{2}{\pi}} \cdot \frac{e^{R-Np}}{R^{3/2}} \tag{17}$$

Note that,

$$\begin{aligned} \tfrac{df(p)}{dp} &\to \frac{e^{R-Np}\left(\frac{dR}{dp}-N\right)R^{3/2} - \frac{3}{2}R^{1/2}\cdot\frac{dR}{dp}\cdot e^{R-Np}}{R^3} \\ &= \tfrac{N}{R^3}\sqrt{\tfrac{2}{\pi}}\left(N(2-4p-R) - \tfrac{3N(1-2p)}{R}\right) < 0 \end{aligned} \tag{18}$$

Therefore, we prove that, as $N \to \infty$, $f(p)$ is a monotonically decreasing function for $\ln N/N < p < 1 - \ln N/N$. ■

Lemma IV.2. *Let $p_c = N^{\varepsilon-1} \ln N$, where $0 < \varepsilon \ll 1$. Then we have $f(p_c) \to 0$ as $N \to \infty$.*

Proof: Note that $0 < \varepsilon \ll 1$, thus we have $p_c \to 0$ and $1 - p_c \to 1$ as $N \to \infty$. Then we obtain that $R_{p_c} \to 2\sqrt{N^\varepsilon \ln N}$. Therefore, we have

$$\begin{aligned} f(p_c) &\to N\sqrt{\frac{2}{\pi}} \cdot \frac{e^{R_{p_c} - Np_c}}{R_{p_c}^{3/2}} = N\sqrt{\frac{2}{\pi}} \cdot \frac{e^{2\sqrt{N^\varepsilon \ln N} - N^\varepsilon \ln N}}{\left(2\sqrt{N^\varepsilon \ln N}\right)^{3/2}} \\ &= \frac{N^{1-3\varepsilon/4}}{2\sqrt{\pi}} \cdot \frac{e^{2\sqrt{N^\varepsilon \ln N} - N^\varepsilon \ln N}}{(\ln N)^{3/4}} = \frac{N^{1-3\varepsilon/4}}{2\sqrt{\pi}} \cdot \frac{e^{2\sqrt{N^\varepsilon \ln N} - N^\varepsilon \ln N}}{(\ln N)^{3/4}} \\ &= \frac{N^{1-3\varepsilon/4}}{2\sqrt{\pi}} \cdot \frac{e^{-\left(\sqrt{N^\varepsilon \ln N} - 1\right)^2 + 1}}{(\ln N)^{3/4}} \end{aligned} \tag{19}$$

Since $\sqrt{N^\varepsilon \ln N} \gg 1$ as $N \to \infty$, we obtain that

$$f(p_c) \to \frac{N^{1-3\varepsilon/4}}{2\sqrt{\pi}} \cdot \frac{e^{-\left(\sqrt{N^\varepsilon \ln N} - 1\right)^2 + 1}}{(\ln N)^{3/4}} \approx \frac{eN^{1-3\varepsilon/4 - N^\varepsilon}}{2\sqrt{\pi}\,(\ln N)^{3/4}} \to 0 \tag{20}$$

The proof is completed. ■

From Lemmas 3.1 and 3.2, it is easy to derive that, for $p_c \le p \le 1 - p_c$, $f(p) \le f(p_c) \to 0$ as $N \to \infty$. Consequently, we obtain the following theorem.

Theorem IV.3. *Let $G_{N,p}$ be a random graph with $N^{\varepsilon-1} \ln N < p < 1 - N^{\varepsilon-1} \ln N$, where $0 < \varepsilon \ll 1$. Then the natural connectivity of $G_{N,p}$ almost surely is*

$$\bar{\lambda} = Np - \ln(N) + o(1) \tag{21}$$

where $o(1) \to 0$ as $N \to \infty$.

From Eq. (20), we know that the natural connectivity of random graphs increases linearly with edge density p given the graph size N. Note that $< k >= Np$; thus, we also observe that the natural connectivity of random graphs increases linearly with the average degree given the graph size N. To verify our result, we simulate 1000 independent $G_{N,p}$ and compute the average natural connectivity for each combination of N and p. In Figure 1, we plot the natural connectivity of random graphs with both numerical results and analytical results. We observe that the numerical results agree well with the analytical results.

V. Comparisons between random graphs and regular graphs

To investigate the effect of randomness and small-world on the network robustness, we compare the natural connectivity of Erdős-Rényi random graphs $G_{N,p}$ with regular ring lattices $RRL_{N,2K}$ and random regular graphs $RRG_{N,2K}$ [47]. We choose $p \approx 2K/N$ and $< k >= 2K$ and thus three types of networks have the same number of vertices and edges. The results are shown in Figure 2. We find that random graphs are always robustness than random regular graphs. However, the curves of regular ring lattices cross those of random graphs; furthermore, random graphs are more robust than regular ring lattices prior to crossings (dense networks), whereas regular ring lattices are more robust than random graphs over crossings (sparse networks). This means that there is a critical graph size N_c, that is as a function of K. For example, for $K = 5$, we find that $N_c \approx 60$.

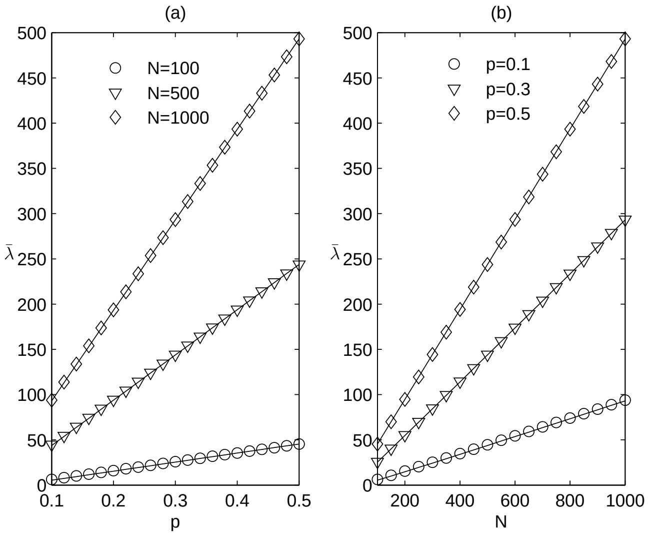

Figure 1. Natural connectivity of random graphs: (a) $\bar{\lambda}$ vs. p with $N = 100$ (circles), 500 (triangles) and 1000 (diamonds); (b) $\bar{\lambda}$ vs. N with $p = 0.1$ (circles), 0.3 (triangles), 0.5 (diamonds). Each quantity is an average over 1000 realizations. The lines represent the corresponding analytical results according to Eq. (20).

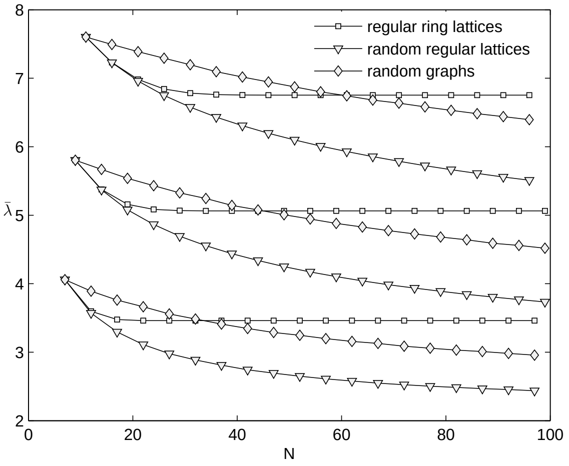

Figure 2. Natural connectivity of random graphs of regular ring lattices $RRL_{N,2K}$ (squares), random regular graphs $RRG_{N,2K}$ (triangles) and random graphs $G_{N,p}$ (diamonds) with the same number of vertices and edges. From bottom to top, the symbols correspond to $K = 3, 4, 5$, respectively. Each quantity is an average over 1000 realizations.

ISBN: 978-1-61208-198-4

For large values of K, we can analytically predict the values of N_c using Eq. (4) and Eq. (20) as follows

$$\ln\left(I_0(\overbrace{2,2,...2}^{K})\right) = Np - \ln(N) = 2K - \ln(N) \tag{22}$$

$$\Rightarrow N_c \approx e^{2K - I_0(\overbrace{2,2,...2}^{K})}$$

The results are shown in Figure 3. Moreover, we also find that there is a critical value p_c or K_c as a function of graph size N. Regular ring lattices are more robust than random graphs when the edge density $p < p_c$, whereas random graphs are more robust than regular ring lattices when the edge density $p > p_c$.

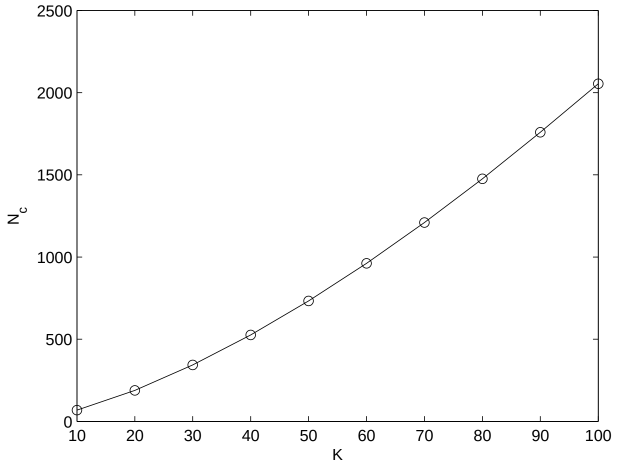

Figure 3. The critical value N_c as a function of graph size K according to Eq. (21).

To explore the critical behaviors of graphs in depth, we randomize regular ring lattices by random rewiring [48] and by random degree-preserving rewiring [49], which leads to random graphs and random regular graphs, respectively. In Figure 4, the natural connectivity is represented as a function of the number of rewiring, starting from regular ring lattices with $N = 30 < N_c$ and $N = 100 > N_c$, where $K = 5$. We find that the natural connectivity decreases during the process of random degree-preserving rewiring and equals to the value of a random regular graph finally. It means that regular ring lattices are more robust than random regular graphs for both $N = 30$ and $N = 100$. The case of random rewiring is more complicated. Different processes of random rewiring for $N = 30$ and $N = 100$ are shown in Figure 4. The natural connectivity increases during the process of random rewiring for $N = 30 < N_c$; however, for $N = 100 > N_c$, the natural connectivity first decreases during the process of random rewiring and then increases during the process of random rewiring; finally, equals to the value of a random graph finally (smaller than the value of a regular ring lattice). It means that randomness increases the robustness of a dense regular ring lattice, but decreases the robustness of a sparse regular ring lattice.

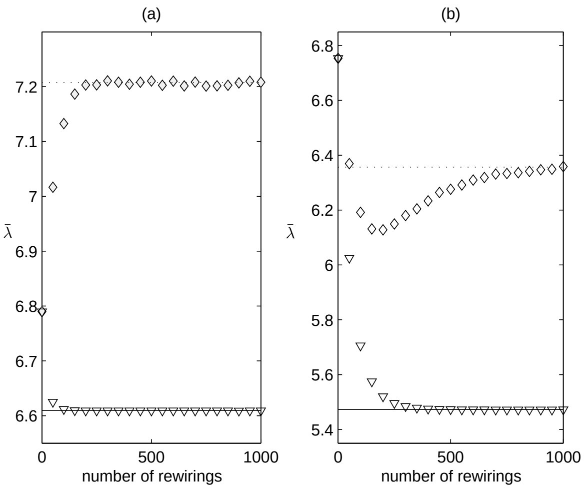

Figure 4. Natural connectivity during the processes of random rewiring (diamonds) and random degree-preserving rewiring (triangles) starting from regular ring lattices with $N = 30$ (a), $N = 100$ (b), where $K = 5$. The solid lines represent the values of random regular graphs and the dashed lines represent the values of random graphs. Each quantity is an average over 1000 realizations.

VI. Conclusion and future work

We have investigated the natural connectivity of Erdős-Rényi random graphs $G_{N,p}$ in this paper. We have presented the spectral density form of natural connectivity and derived the natural connectivity of random graphs analytically using the Wigner's semicircle law. In addition, we have shown that the natural connectivity of random graphs increases linearly with edge density p given a large graph size N. The analytical results agree with the numerical results very well.

We have compared the natural connectivity of random graphs $G_{N,p}$ with regular ring lattices $RRL_{N,2K}$ and random regular graphs $RRG_{N,2K}$ with the same number of vertices and edges. We have shown that random graphs are more robust than random regular graphs; however the relationship between random graphs and regular ring lattices depends on the graph size N and the edge density p or the average degree $< k >$. We have observed that the critical value N_c as a function of K, and the critical value p_c and K_c as a function of graph size N, which can be predicted by our analytical results. We have explored the critical behavior by random rewiring from regular ring lattices. We have shown that randomness increases the robustness of a dense regular ring lattice, but decreases the robustness of a sparse regular ring lattice. Our results will be of great theoretical and practical significance to the network robustness design and optimization.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant Nos 70771111, 60904065, 71031007, 71171195 and 11101429, and the Specialized Research Fund for the Doctoral Program of Higher Education under Grant No 20094307120001.

REFERENCES

[1] A. Vázquez, R. Pastor-Satorras, and A. Vespignani, “Large-scale topological and dynamical properties of the internet,” *Phys. Rev. E*, vol. 65, no. 6, p. 066130, 2002.

[2] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabási, “The large-scale organization of metabolic networks,” *Nature*, vol. 407, no. 6804, pp. 651–654, 2000.

[3] M. Rosas-Casals, S. Valverde, and R. V. Sole, “Topological vulnerability of the european power grid under errors and attacks,” *Int. J. Bifurcat. Chaos*, vol. 17, no. 7, pp. 2465–2475, 2007.

[4] H. P. Thadakamalla, U. N. Raghavan, S. Kumara, and R. Albert, “Survivability of multiagent-based supply networks: A topological perspective,” *IEEE Intell. Syst.*, vol. 19, no. 5, pp. 24–31, 2004.

[5] F. Xie and D. Levinson, “Measuring the structure of road networks,” *Geogr. Anal.*, vol. 39, no. 3, pp. 336–356, 2007.

[6] M. A. Serrano and M. Boguñá, “Topology of the world trade web,” *Phys. Rev. E*, vol. 68, no. 1, p. 015101, 2003.

[7] R. Albert and A. L. Barabási, “Statistical mechanics of complex networks,” *Rev. Mod. Phys.*, vol. 74, no. 1, pp. 47–97, 2002.

[8] M. E. J. Newman, “The structure and function of complex networks,” *Siam Rev.*, vol. 45, pp. 167–256, 2003.

[9] L. A. N. Amaral and B. Uzzi, “Complex systems - a new paradigm for the integrative study of management, physical, and technological systems,” *Management Sci.*, vol. 53, no. 7, pp. 1033–1035, 2007.

[10] H. Whitney, “Congruent graphs and the connectivity of graphs,” *Am. J. Math.*, vol. 54, no. 1, pp. 150–168, 1932.

[11] G. Bauer and G. Bolch, “Analytical approach to discrete optimization of queuing-networks,” *Comput. Commun.*, vol. 13, no. 8, pp. 494–502, 1990.

[12] F. Harary, “Conditional connectivity,” *Networks*, vol. 13, no. 3, pp. 347–357, 1983.

[13] A. H. Esfahanian and S. L. Hakimi, “On computing a conditional edge-connectivity of a graph,” *Inform. Process. Lett.*, vol. 27, no. 4, pp. 195–199, 1988.

[14] M. S. Krishnamoorthy and B. Krishnamurthy, “Fault diameter of interconnection networks,” *Comput. Math. Appl.*, vol. 13, no. 5-6, pp. 577–582, 1987.

[15] V. Chvátal, “Tough graphs and hamiltonian circuits,” *Discr. Math.*, vol. 5, pp. 215–228, 1973.

[16] H. A. Jung, “Class of posets and corresponding comparability graphs,” *J. Comb. Theory B*, vol. 24, no. 2, pp. 125–133, 1978.

[17] M. Cozzen, D. Moazzami, and S. Stueckle, “The tenacity of a graph,” in *Seventh International Conference on the Theory and Applications of Graphs*. New York: Wiley, 1995, pp. 1111–1122.

[18] N. Alon, “Eigenvalues and expanders,” *Combinatorica*, vol. 6, no. 2, pp. 83–96, 1986.

[19] B. Mohar, “Isoperimetric number of graphs,” *J. Comb. Theory B*, vol. 47, pp. 274–291, 1989.

[20] M. Fiedler, “Algebraic connectivity of graphs,” *Czech. Math. J.*, vol. 23, pp. 298–305, 1973.

[21] B. Bollobás, *Random Graphs*. New York: Academic Press, 1985.

[22] R. Albert, H. Jeong, and A. L. Barabasi, “Error and attack tolerance of complex networks,” *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[23] V. Latora and M. Marchiori, “Efficient behavior of small-world networks,” *Phys Rev Lett*, vol. 87, p. 198701, 2001.

[24] G. Siganos, S. L. Tauro, and M. Faloutsos, “Jellyfish: a conceptual model for the as internet topology,” *J Commun Netw*, vol. 8, no. 3, pp. 339–350, 2006.

[25] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin, “Breakdown of the internet under intentional attack,” *Phys. Rev. Lett.*, vol. 86, no. 16, pp. 3682–3685, 2001.

[26] ——, “Breakdown of the internet under intentional attack,” *Phys. Rev. Lett.*, vol. 86, no. 16, pp. 3682–3685, 2001.

[27] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, “Network robustness and fragility: percolation on random graphs,” *Phys. Rev. Lett.*, vol. 85, no. 25, pp. 5468–5471, 2000.

[28] J. Wu, H. Z. Deng, Y. J. Tan, and Y. Li, “A robustness model of complex networks with tunable attack information parameter,” *Chin. Phys. Lett.*, vol. 24, no. 7, pp. 2138–2141, 2007.

[29] J. Wu, Y. J. Tan, H. Z. Deng, and D. Z. Zhu, “Vulnerability of complex networks under intentional attack with incomplete information,” *J. Phys. A*, vol. 40, no. 11, pp. 2665–2671, 2007.

[30] J. Wu, M. Barahona, Y. J. Tan, and H. Z. Deng, “Natural connectivity of complex networks,” *Chin. Phys. Lett.*, vol. 27, no. 7, p. 078902, 2010.

[31] E. Estrada, “Characterization of 3d molecular structure,” *Chem. Phys. Lett.*, vol. 319, no. 5-6, pp. 713–718, 2000.

[32] E. Estrada and J. A. Rodrłguez-Velzquez, “Spectral measures of bipartivity in complex networks,” *Phys. Rev. E*, vol. 72, no. 4, p. 046105, 2005.

[33] ——, “Subgraph centrality in complex networks,” *Phys. Rev. E*, vol. 71, no. 5, p. 056103, 2005.

[34] E. Estrada, “Network robustness to targeted attacks. the interplay of expansibility and degree distribution,” *Eur. Phys. J. B*, vol. 52, no. 4, pp. 563–574, 2006, times Cited: 5.

[35] ——, “Spectral scaling and good expansion properties in complex networks,” *Europhys. Lett.*, vol. 73, no. 4, pp. 649–655, 2006.

[36] Y. L. Shang, “Local natural connectivity in complex networks,” *Chinese Physics Letters*, vol. 28, no. 6, p. 068903, 2011.

[37] ——, “Perturbation results for the estrada index in weighted networks,” *Journal of Physics A*, vol. 44, no. 7, 2011, 075003.

[38] P. Zhang and Q. S. Ma, “A method of evaluating robustness of more-electrical-aircraft power system based on natural connectivity,” in *6th IEEE Conference on Industrial Electronics and pplications*. Beijing: IEEE, 2011, pp. 158–160.

[39] J. Wu, M. Barahona, Y. J. Tan, and H. Z. Deng, “Robustness of regular ring lattices based on natural connectivity,” *Int. J. Syst. Sci.*, vol. 42, no. 7, pp. 1085–1092, 2011.

[40] P. Erdős and A. Rényi, “On random graphs,” *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[41] ——, “On the evolution of random graphs,” *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[42] F. Chung, L. Y. Lu, and V. Vu, “Spectra of random graphs with given expected degrees,” *Proc. Nat. Acad. Sci. U.S.A.*, vol. 100, no. 11, pp. 6313–6318, 2003.

[43] M. Krivelevich and B. Sudakov, “The largest eigenvalue of sparse random graphs,” *Combin. Probab. Comput.*, vol. 12, no. 1, pp. 61–72, 2003.

[44] E. Wigner, “Characteristic vectors of bordered matrices with infinite dimensions,” *Ann. of Math.*, vol. 62, no. 3, pp. 548–564, 1955.

[45] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1972.

[46] G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists*, 6th ed. San Diego: Academic Press, 2005.

[47] A. Steger and N. C. Wormald, “Generating random regular graphs quickly,” *Combin. Probab. Comput.*, vol. 8, no. 4, pp. 377–396, 1999.

[48] D. J. Watts and S. H. Strogatz, “Collective dynamics of ’small-world’ networks,” *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[49] M. E. J. Newman, “Assortative mixing in networks,” *Phys. Rev. Lett.*, vol. 89, no. 20, p. 20871, 2002.