Reliability Estimation for Large Distributed Software Systems

Alberto Avritzer, Flávio P. Duarte
Siemens Corporate Research
alberto.avritzer@siemens.com, flavio.p.duarte@gmail.com

Rosa Maria Meri Leão, Edmundo de Souza e Silva
Universidade Federal do Rio de Janeiro
rosam@land.ufrj.br, edmundo@land.ufrj.br

Michal Cohen, David Costello
Siemens Transportation Systems
michal.cohen@siemens.com, david.costello@siemens.com

Abstract

In this paper, we present our experience to estimate the reliability of a large distributed system composed of several hundred points of presence. The system's reliability metric is required by contract to be obtained. A simple approach is presented to accurately approximate the desired metrics.

1 Introduction

In large industrial software development efforts, it is usually important to track software quality improvement during system test. Reliability growth models have been commonly used to account for quality improvement during system test efforts, as fault detection and removal takes place. An example of the application of reliability growth models to the estimation of the reliability metrics of telecommunication switches was presented in [9]. However, once the software is deployed in production, reliability is usually estimated by observing actual failures and estimating the overall system failure rate [7]. When the system under study is composed of very reliable nodes, failures are likely to be rare events, and the failure sample size is usually small.

In this paper, we present an experience report of the reliability estimation of a large industrial distributed system. For this system, which we call the system under study, we were required by contract to estimate the reliability of a large distributed network of nodes. As a result, we have developed an approach that can be used to derive an approximation of the reliability for the large network of distributed nodes. The important system failure parameters for this effort were identified as the system mission time and the service impact of failure on the system nodes. In addition, the developed approach had to account for vendor reliability data and field-measured failure data. The approximation of the reliability was used to demonstrate that the deployed network of nodes meets the contracted reliability objective.

In [7] important issues related to reliability metrics interpretation by designers, project managers and other stakeholders within a project organization were discussed. The authors showed how difference in interpretation of reliability metrics could lead to increased

costs to the organization due to misunderstandings of the meaning of Mean Time to Failure and its confusion with measurements of failure rates that were based on small samples. Therefore, the authors recommended the use of statistically-proven approaches for reliability modeling. In [7], it was shown that the shape of the reliability function is usually a bathtub curve as shown in Figure 1. Therefore, it is important for data analysis approaches to statistically validate that the nodes under test are after infant-mortality phase and before wear-out phase. It was also reported in [7] that a twelve-month aggregate average should be used to estimate the constant hazard rate. As the testing period in the case study presented in this paper was three months, we used a combination of vendor data and failure data to ensure the nodes under test were in the constant failure rate phase and to adjust for error introduced by the short three-month reliability period for data collection.

In the large distributed software environment under study in this paper, service is provided to applications at several hundred locations, which we call the system points of presence (*POPs*). In addition, alternate routes are provided between important *POPs*. The system under study was designed to be fault-tolerant to certain node faults. If such faults are repaired in time, there would be no service affecting failure associated with these faults, as service would continue to be provided to the associated *POPs* through alternate routes. Even if the fault manifests itself as a failure at the service boundary, a small fraction of the available *POPs* would be usually affected.

Network reliability analysis problems have been
shown to be *NP-Hard* [5]. Monte Carlo simulation methods [6] and Artificial Neural Networks (*ANN*) [11] have been used to estimate large networks reliability. In this paper, we take advantage of the structure of the distributed system under study and apply *Transient Markov Chain Analysis* [14, 13] to derive an approximation of the reliability of a very large distributed system composed of several hundred points of presence (*POPs*).

Section 2 presents the distributed system under study and Section 3 presents the Markov model we use in this work to estimate the system reliability. The equation used to compute the approximation of the system reliability is introduced in Section 4. Section 5 presents an analysis of the approximation error. The summary of the proposed reliability estimation approach is introduced in Section 6. In Section 7, we present our conclusions and suggestions for future research.

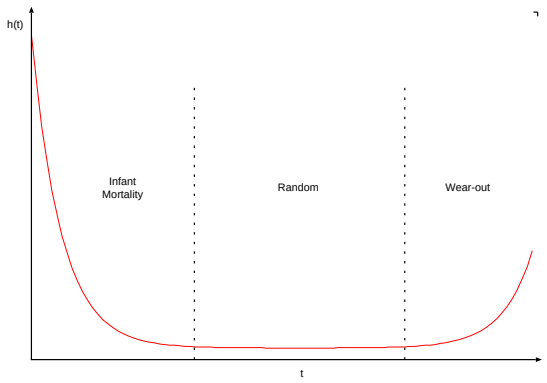

Figure 1: Hazard Rate Function

2 System under Study

For the reliability estimation purpose, the node faults to be considered are those faults that propagate to the service boundary. These faults are therefore classified as service-affecting failures. Because the system under study was provisioned with alternate routes to points of presence (*POP*), a *POP* is said to be service-impacted if more than one node failure leads to a service outage at the *POP*.

Table 1 shows how many *POPs* belong to each area group(AG). The node failure to *POP* impact mapping in Table 1 will be used in Equation 1 to weight each failure state probability by the number of *POPs* that were service-affected by the associated equipment failure.

The next step in the reliability-estimation process is to compute the number of *POPs* affected by an area outage. Table 2 shows the relationship between node failures and the affected AGs. Tables 1 and 2 were derived from the system topology. The system topology is

not described to protect the customer's intellectual property.

The data contained in Tables 1 and 2 show how many *POPs* are under control of a set of nodes. For example, if node equipment labeled NOD13 fails, area group 95 will be unreachable, with service impact to one *POP*. In contrast, if the pair of nodes NOD20 and NOD23 fails, two area groups (12 and 39) will be affected by the outage, with service impact to 18 *POPs*.

Table 1: Number of *POPs* per Area Group

AG	POPs	AG	POPs	AG	POPs
1	7	13	11	45	1
2	10	14	5	46	2
3	9	15	5	51	1
4	4	16	8	56	6
5	3	17	7	62	2
6	3	18	8	67	5
7	10	19	9	92	1
8	8	20	12	93	1
9	13	34	1	94	1
10	8	39	12	95	1
11	8	43	11	96	1
12	6	44	1	97	1
				98	1

Table 2: Node Failure Impact on the Area Groups (AGs)

Node	Aff. AGs	Node	Aff. AGs
NOD02, NOD18	1	NOD08, NOD20	10, 11
NOD09, NOD20	9	NOD09, NOD02	7, 8
NOD02, NOD03	56	NOD03, NOD20	20
NOD03, NOD22	62	NOD03, NOD18	51
NOD07, NOD17	46, 92	NOD20, NOD23	12, 39
NOD20, NOD24	13	NOD23, NOD24	14
NOD23, NOD07	16, 17	NOD17	92
NOD07, NOD15	93, 45	NOD05, NOD17	34, 92
NOD05, NOD15	44, 93	NOD05, NOD19	18
NOD13	95	NOD05, NOD10	43
NOD14	94	NOD24, NOD11	15
NOD24, NOD21	6	NOD21, NOD22	5
NOD12	98	NOD01, NOD18	67
NOD01, NOD22	4	NOD04	96
NOD10, NOD18	2	NOD06, NOD19	19
NOD16	97	NOD10, NOD19	3

3 The Markov Model

In this section, we present the discrete state continuous time Markov modeling approach we used to evaluate the impact of failures on the system reliability.

We define as the Markovian state, $S(t)$, the list of nodes that have failed up to time t. For example, if by time t, nodes $NOD10, NOD19$ have failed, the Markov model will be in state, $S(t) = (NOD10, NOD19)$.

Therefore, a node failure is represented in the Markov model as a state transition. The transition rate between states is represented by the failure rate of the node responsible for the specific transition. The initial state at time $t = 0$ represents the situation where all the nodes are fully operational.

For a two-node system, the Markov model of Figure 2 has four states, while for a three-node system, the Markov model of Figure 4 has eight states.

For a large system with 72 nodes, the full Markov model would have $4.4E21$ states. Therefore, we use in this paper a a Model simplification, shown in Figure 3, that results in an approximation of the estimated reliability. In our approximation, we include in the Markov model only those states that cause a service outage after the occurrence of two node failures. In the model of Figure 3, we represent all the node failures that cause a service outage after the occurrence of two failures. As a result, instead of representing $4.4E21$ states that would be required for the full Markov model, the Markov model, for 72 nodes, contains only 109 states. This is an approximation of the reliability because we have discarded from the model all states that are reachable by two-step node failures and do not represent a service outage condition. In contrast, we have included all states that are reachable by two-step failure transitions and represent a service outage.

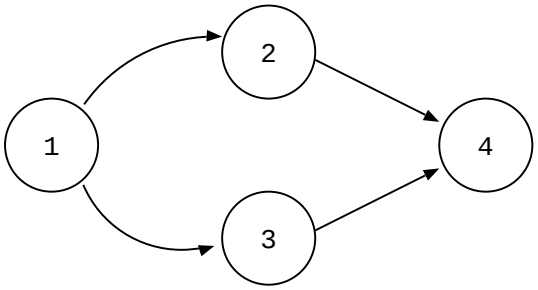

Figure 2: Markov Failure Model

A failure that causes a *POP* group to become unreachable is represented in our Markov model by an end state (sink state). A sink state is defined as a state that has only arriving transitions to it but no departing transitions.

Figure 2 displays the two-node system model that illustrates the above concepts. *State 1* in Figure 2 represents the system with all components working. *State 2* represents a single equipment failure. Similar to *state 2*, *state 3* also represents an equipment failure, but a different one. *State 4*, which is a sink state, represents a state where both nodes have failed.

Figure 3 displays the Markov chain that represents the approximation for a system with 72 nodes. The central state, labeled "Ok", models the system when all components are working properly. The states surrounding the "Ok" state represent the situation that occurs when one node failure has occurred, but no *POP* has become unreachable. The remaining states, i.e., the sink states, represent the situation where more than one node failure has occurred, and at least one *POP* has become isolated.

The recommended approach for the solution of Markovian models with sink states is to use the transient solution method [15], which computes the state probability distribution for a certain mission time. In our reliability estimation approach, we used the Tangram-II [14] tool transient solution method.
Tangram-II is a tool for analysis and simulation. It has several different solutions techniques implemented for both stationary and transient analysis of Markovian models.

The failure rate values used are a combination of vendor's data-sheets and the failure rates estimated from the data collected during the system reliability testing effort. For example, in the cases where no failures would be observed during the testing phase, the failure rate used will be the failure rate provided by the node equipment vendor. The computation of failure rates using the vendor's data and measured failure data is presented in Section 6.

4 Reliability Estimation

In [3], a domain-based reliability estimation approach for systems that could be modeled as Markov-chains was introduced. The approach introduced in [3] used a distance function to incorporate service impact into the reliability metric. In [4], the approach introduced in [3] was applied to estimate the reliability for a large industrial rule-based system. In this paper, we extend the approach introduced in [3] to estimate the reliability of a large industrial system with several hundred *POPs*.

For this system reliability estimation effort, we have defined a reliability estimation function that weights the probability of being at a specific failure state i by the fraction of *POPs* impacted by a service outage at state i. Equation 1 shows the proposed reliability estimation function:

$$R(t) = 1 - \sum_{i} \left(P_t(i) * \frac{N_{fail}(i)}{N_{total}} \right) \quad (1)$$

where $N_{fail}(i)$ is the number of *POPs* that are impacted by the service outage represented by state i, N_{total} is the total number of *POPs*, and $P_t(i)$ is the probability of occurrence of the service outage represented by state i, for mission time t.

5 Estimating the Reliability Estimation Error

The approach presented in Section 3 for Markov chain modeling of large distributed systems was to include in the Markov model only those states that are associated with a service

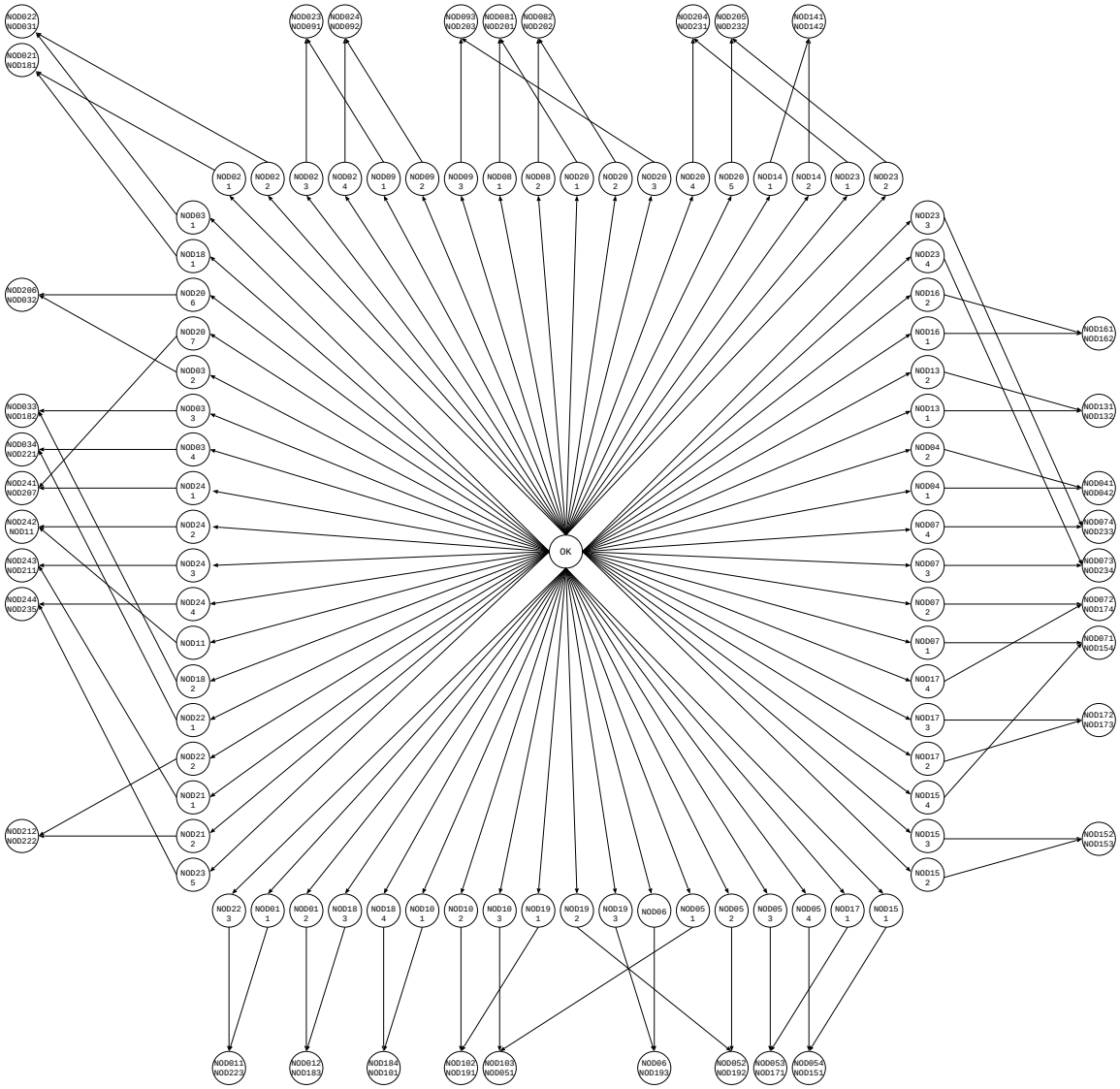

Figure 3: Markov Model representing Seventy-Two Node System

outage caused by two-step node failures. In our Markov modeling approximation, we took advantage of the structure of our system network topology to build the Markov chain model shown in Figure 3.

In this section, we estimate the error introduced by discarding states that represent node failures that do not cause a service outage. We compare the simple model of Figure 2, where the states that were not in the direct path of a service outage were discarded, with the Markov model shown in Figure 4, where each state contains three variables representing the state of nodes A, B, and C respectively. In the Markov model of Figure 4, 1 represents the situation where the associated node is fully operational, and 0 represents the situation where the associated node has failed. For example, the state (1,0,1) represents the situation where nodes A and C are fully operational, while node B has failed. As in the simple model, the system contains a service outage, when both nodes A and

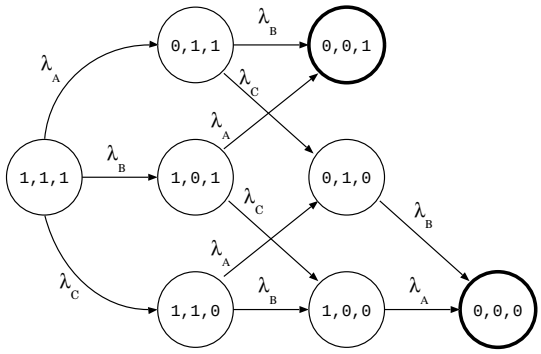

Figure 4: Markov Model with Three Nodes

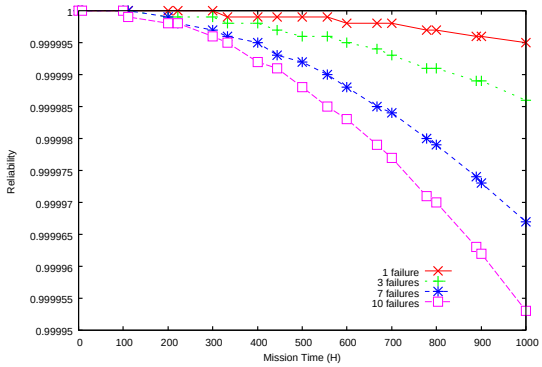

Figure 5: Two Node Reliability versus Mission Time (h) for 1, 3, 7, 10 POP

B have failed; so, in this model, the sink states that represent a service outage are (0,0,0) and (0,0,1). These states are marked with bold circles in Figure 4.

Figure 5 presents the graph of the Reliability metric versus Mission Time for the approximation approach shown in Figure 2. The two-node model is obtained from the three-node model by discarding states reached by two-step node failures that do not represent service outages. We have computed these plots using equation 1 to account for the number of *POPs* impacted by the service outage.

For the purpose of demonstrating the impact of service outages on the reliability metric, we have used the values of 1, 3, 7, and 10 for the number of affected *POPs* in Figure 5. The actual number of *POPs* affected by a node failure in the system under study is shown in Table 1.

We have also computed the reliability for the same 4 outage scenarios, but for the three-node model to estimate the error introduced by our approximation. For example, using 10 *POP* service outage impact and 1000 hour mission time, the approximation on the reliability was estimated as 0.999955, while the actual reliability was estimated as 0.999997.

Figure 7 shows the reliability approximation for the number of affected *POPs* values of 1, 3, 7 and 10, for the large industrial system with seventy-two nodes. We can see from Figure 7 that the reliability objective of 0.999 is met for the evaluated values of affected *POPS* up to 600 hours of mission time. Therefore, for industrial contracts where there is a requirement to demonstrate a reliability objective, for a given range of mission times, the approximation presented in this paper is very practical. In our approach, these approximations were computed efficiently, taking about 5-10 seconds of CPU time on a Dell 830 Laptop. The errors introduced by the approximation of the reliability were not significant to the industrial application under study.

6 Summary of Approach

In this section, we summarize the reliability estimation steps presented in the previous sections.

The overall algorithm is as follows:

1. Using Equation 2 estimate the equipment failure rate from: (a) estimates from the vendor specification sheet and; (b) the measured data equipment failure rate.

$$E_f = \frac{nf_v + nf_m}{t_v + t_m} \tag{2}$$

where E_f is the estimated equipment failure rate, nf_v is the number of failures measured by the vendor during its estimation of the reliability, nf_m is the number of failures measured during the data collection phase, t_v is the time used to collect nf_v, and t_m is the time used to measure nf_m.

The new MTBF will be:

$$MTBF = \frac{1}{E_f} \tag{3}$$

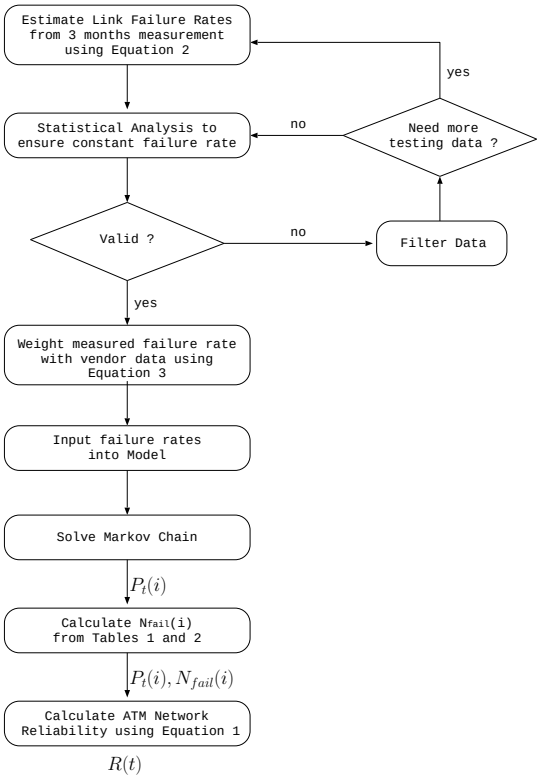

Figure 6: Algorithm to Estimate Reliability

2. Input failure rate from Step 2 into Markov model from Figure 3. After the data collection period, scripts will extract the failure rate computed per Equation 2 for all equipments.

3. Solve Markov model to obtain $P_t(i)$ using mission time t=8 hours. The Tangram-II tool will analytically solve the model using a transient technique for a mission time equal to 8 hours; the output will be the set of state probabilities ($P_8(i)$).

4. Obtain $N(i)$ from Tables 1 and 2. Tables 1 and 2 will be combined to calculate the number of *POPs* that were service-affected due to a specific failure (i).

5. Calculate reliability using Equation 1.

Figure 6 presents the algorithm used to calculate the system reliability. The statistical analysis uses a linear regression test to validate

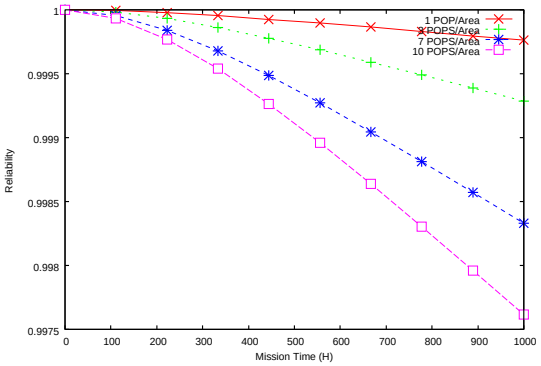

Figure 7: Seventy-Two Node System Reliability versus Mission Time (h) for 1, 3, 7, 10 POP

the constant failure rate hypothesis. If the test for constant failure rate fails, the algorithm iterates by filtering the data to eliminate outliers. If a determination is made that further test is required, the whole procedure is repeated.

7 Conclusions

We have presented an experience report of the derivation of a computationally-efficient approximation of the reliability of a large industrial distributed system that takes into account the impact of node failures on service. We have defined a reliability estimation function that weights the probability of being at a specific failure state by the fraction of points of presence impacted by a specific service outage.

In addition, we have defined a Markov model that captures all two-step transitions into service-affecting states and discards all two-step or longer transitions into states that are not service-affecting. Our approach takes into account the structure of the system network topology to derive a simple Markov chain that is able to represent all two-step service outages.

We have estimated the error of our approach by building a two-node approximation for a three-node system, and a three-node Markov model that captures the full state transitions of a three-node system. In addition, we have computed the approximation of the reliability for the large, seventy-two node industrial sys-

tem under study. The approximation shows that for the range of parameters studied the large industrial system will meet contracted reliability objectives.

Therefore, it is our experience that for large industrial systems, where there is a requirement to demonstrate a reliability objective for a given range of mission times, the approximation presented in this paper is very practical, as the solution of full state problems is combinatorial with the number of states.

We have had limited experience with assessing reliability for large industrial systems, as we have applied our methodology to only one large system. Therefore, we would like to apply our methodology to additional industrial systems to further validate its practicality.

In future research, we would like to develop lower bounds for the reliability by capturing additional situations where some nodes have failed without having an impact at the service boundary.

About the Authors

Alberto Avritzer received a Ph.D. in Computer Science from the University of California, Los Angeles, an M.Sc. in Computer Science for the Federal University of Minas Gerais, Brazil, and the B.Sc. in Computer Engineering from the Technion, Israel Institute of Technology. He is currently a senior member of the technical staff at Siemens Corporate Research, Princeton, New Jersey. He spent the summer of 1987 at IBM Research at Yorktown Heigths. His research interests are in software engineering, particularly software testing and reliability, real-time systems, and performance modeling, and has published several papers in those areas. He is a member of ACM SIGSOFT, and IEEE.

Flávio P. Duarte received his B.Sc. degree in Computer Science from UFRJ in 1997. During 1998 Flavio worked for Modulo S/A to secure the Brazilian Federal Election. In 2003 Flavio received his M.Sc. degree in System Engineering with emphasis in Computer Networking from COPPE/UFRJ. His research interests include performance analysis of computer systems, and modeling and simulation of computer networks.

Rosa M. M. Leão received the B.Sc. degree in Computer Science from the Federal University of Rio de Janeiro in 1983, the M.Sc. degree in Computer Science from PUC-Rio in 1990, and the Ph.D. degree in Computer Science from the Paul Sabatier University (LAAS) in 1994. Currently she is an Associate Professor in the Systems Engineering and Computer Science Department at the Federal University of Rio de Janeiro. Her research interests include computer networks, computer and communication systems modeling, performance evaluation, and multimedia systems.

Edmundo de Souza e Silva received the B.Sc. and M.Sc. degrees in electrical engineering, both from Pontifical Catholic University of Rio de Janeiro (PUC/RJ), and the Ph.D. degree in computer science from the University of California, Los Angeles in 1984. Currently he is a professor at the Federal University of Rio de Janeiro, COPPE and Computer Science Department. His areas of interest include the modeling and analysis of computer systems and computer communication multimedia networks.

Michal Cohen received a B.Sc. in Math and Computer Engineering from the Ben Gurion University, Israel. She is currently working as a Network Integration Manager at Siemens Transportation Systems. She has over 8 Years of Network Communication experience working as a Network Integration Manager at the NYCT SONET/ATM Communications Network System (SACNS) Project. In addition, she has also worked as a senior Software Designer at the Signaling and Connection Management Group participating in the architectural design and implementation processes of various communication products.

David Costello is the Project Director of three large communications projects for Siemens Transportation Systems in New York City. These are three of several projects Siemens has in New York City designed improve the subways infrastructure. David joined Siemens in 2001 as an Integration Manager with Siemens Communications; this division is one of the leading global players of the commu-

nications industry. David has managed a team of engineers to design, integrate and commission a new communications network for New York City Transit. This network is the largest transit communications project in the world and will be the basis for all future communications and applications within the New York subway system. David received a Bachelors of Science degree in Electrical Engineering from Polytechnic University and a Masters of Business Administration in Finance from the New York University, Stern School of Business.

References

[1] A. Avizienis and D. E. Ball. On the achievement of a highly dependable and fault-tolerant air traffic control system. IEEE Computer, 1987, pp 84-90.

[2] A. Avritzer and B. Larson. Load testing software using deterministic state testing. In T. Ostrand and E.J.Weyuker, editors, Proceedings of the 1993, International Symposium on Software Testing and Analysis(ISSTA). ACM Press, June 1993, pp. 82-88.

[3] A. Avritzer and E. J. Weyuker. *The Automatic Generation of Load Test Suites and the Assessment of the Resulting Software.* IEEE Trans. on Software Engineering, Sept 1995, pp. 705-716.

[4] A. Avritzer, J. Ros and E. J. Weyuker. Reliability Testing of Rule-Based Systems. *IEEE Software*, September 1996, pp. 76–82.

[5] M. O. Ball. Computational Complexity of Network Reliability Analysis: An Overview. *IEEE Trans. on Reliability*, Vol. R-35, NO.3, August, 1986, pp. 230-238.

[6] S-M. Huang and Q. Wu and S-C. Tsai. A Monte Carlo Method for Estimating the Extended All-Terminal Reliability. *Fourth International Conference on Networking and Services, 2008* March 16-21, 2008, pp. 122-127.

[7] I. James and J. Marshal, M. Evans, B. Newman. Reliability Metrics and the REMM Model. IEEE RAMS 2004, pages 474-479.

[8] J. Jones and J. Hayes. Estimation of System Reliability Using a 'Non-Constant Failure Rate' Model. *IEEE Transactions on Reliability*, Vol. 50, No. 3, Sept 2001.

[9] K. Kanoun and M. R. Martini and J. M. de Souza. A Method for Software Reliability Analysis and Prediction Application to the TROPICO-R Switching System. *IEEE Transactions on Software Engineering*, Vol. 17, No. 4, April 1991.

[10] J. D. Musa, A. Iannino, and K. Okumoto *Software Reliability: Measurement, Prediction, Application* McGraw-Hill Book Company, 1987.

[11] F. Altiparmak and B. Dengitz and A. E. Smith. *Reliability Estimation of Computer Communication Networks: ANN Models.* Proc. of the Eighth International Symposium on Computers and Communication, ISCC'03. Turkey, June, 2003.

[12] J. H. Wensley. Fault-tolerant computers ensure reliable industrial controls. August Systems Incorporated, June 1981, OR 97302.

[13] E. de Souza e Silva and R. M. M. Leão and M. C. Diniz, Transient analysis applied to traffic modeling. *Performance Evaluation Review.* Vol.28:(4), 2001, pp 14-16.

[14] E. de Souza e Silva, R. M. M. Leão, Richard R. Muntz, Ana P. C. da Silva, Antonio A. de A. Rocha, Flávio P. Duarte, Fernando J. S. Filho, Guilherme D. G. Jaime, Modeling, analysis, measurement and experimentation with the Tangram-II integrated environment. In *Proc. of Int. Conf. on Performance Evaluation Methodologies and Tools (ValueTools'06)*, 2006.

[15] E. de Souza e Silva and H.R. Gail. Transient Solutions for Markov Chains. Computational Probability, W. Grassmann, editor, Kluwer Academic Publishers, 2000, pp. 43-81.