Modeling Correlated Alarms in Network Management Systems *

Livio Ricciulli and Nachum Shacham
Computer Science Laboratory
SRI International
Menlo Park, CA 94025-3493
e-mail:{livio,shacham}@csl.sri.com

Keywords: Alarm modeling, Network management, Alarm correlation, Network modeling, Network simulation.

Abstract

We introduce a new model for describing the behavior of complex interconnection networks in the presence of faults. With our model, it is possible to obtain the detailed behavior of a faulty telecommunication network by specifying some parameters that describe its physical characteristics. We describe our model, we show how it can be applied to study some real network management structures, and we give some experimental data to evaluate its relation to existing standard analytical models.

1. Introduction

Modern telecommunication networks have grown in complexity beyond the point where they can be managed by a manual process alone. This complexity gave rise to automated network management (NM) systems that interact with network elements (NEs) to gain data about their behavior and performance, and to control their behavior by setting operational parameters. Several protocols have been proposed for the interaction between NEs and NM systems. At present, the Simple Network Management Protocol (SNMP) is the most accepted and deployed. In addition to the development of such protocols, much effort was devoted to the development of an NM user interface. Multiple NM systems are now being offered commercially from such vendors as Hewlett-Packard and Sun Microsystems[1].

Most of the work thus far on NM has focused on the protocols for exchanging NM information and on algorithms for the analysis of this information. The objective of these protocols and algorithms is to maintain high performance and correct behavior of the managed data network. However, the performance of the NM system itself, which has a strong effect on the data network, has received little attention.

The performance of an NM system depends critically on the volume and temporal characteristics of the control traffic it generates, the capacity of the channels allocated to carry the NM traffic, and the architecture of the NM system processing elements. The resources of the NM system must be designed with enough capacity to carry and process the generated control traffic. Insufficient NM resources result in reduced performance of the NM system itself, which in turn adversely affects the performance and behavior of the data network it manages. The design of an NM system must be based on a model of the generated traffic and its flow, so that potential bottlenecks can be identified, and their effect on delay in monitoring data traffic ascertained. The study of the generated traffic is a critical element in NM evaluation and it must result in an understanding of the temporal statistics and geographical distribution of the generated NM traffic. Moreover, it is common for an NM system to generate excessive traffic that carries little value but unless filtered at early stages consumes link capacity and nodal processing resources.

In this paper we focus on the NM traffic-generation process. NM traffic can be partitioned into predictable and random components. The predictable segment comprises such messages as status updates that are sent by network elements periodically or in response to queries by NM

*This work was partially supported by KDD

[1]All product and company names mentioned in this document are the trademarks of their respective holders.

managers. The random component includes traffic that is sent unsolicited by NEs when certain conditions are met (e.g., a trap is set off). Alarms, which are messages that indicate troubled conditions, constitute a large portion of the random traffic.

Alarms, which can be generated by any NE, are usually triggered by external events such as power failure, local congestion, loss of synchronization, or cable breakup. Any such event is likely to be felt by more than one NE, and therefore to trigger multiple alarms. Since all these alarm reports stem from the same problem, this effect is known as alarm *correlation*, and is a major cause for surge in NM traffic and the increased congestion it causes.

Correlated alarms contain highly redundant data that alert the NM system about the ultimate cause of the alarm. Filtering techniques and fault-isolation expert systems are being developed in an attempt to deduce the ultimate cause for the alarms. Most of the studies done on this increasingly important aspect of NM systems ignore the effect of the traffic surge on the network's ability to deliver the alarms and the analysis processors to process them. To ascertain the sensitivity of NM systems to control traffic congestion, one must gain a better understanding of the temporal and geographical distribution of the generated traffic.

The complexity of the traffic-generation process limits the utility of analytical modeling to quite simple scenarios. A simulation model can incorporate more realistic cases, thereby leading to more accurate results. An analytical model, however, can be useful in confirming the simulation results. This approach was taken in our work.

We instantiate our alarm-generation model on a general NM system that abstracts the structure of modern NM systems. Our model takes as its input the topology of the NM system, fault occurrence processes, probabilistic correlation interdependency among NEs, communication speed, processing speed, and storage parameters. Because of its high level of detail, our model is best suited for simulation (although for simple NM systems it can be used for analytical studies). The simulation of our model generates NM traffic according to generalized parameters and measures NM throughput and end-to-end delay of NM traffic.

In the following sections, we present the details of the NM system and alarm-generation models, describe the simulation methodology and its user interface, and discuss performance results that represent the simulator output over multiple test runs. Section 2 discusses previous work in the area of correlated alarms in NM systems. Section 3 presents our NM traffic models, Section 4 presents some simulation experiments that evaluate performance estimation results obtained with our modeling technique, and Section 5 concludes the discussion by summarizing our results and laying down the foundations for further work.

2. Related Work

Management of modern telecommunication networks is a challenging task because these systems are usually composed by a heterogeneous mix of hardware modules, all of which are highly dependent on each other. Network management therefore needs to abstract the behavior of all the different components of a network and provide the means to perform, in a resilient way, critical and complex management operations. Much research today is dedicated to advancing the knowledge and expertise in this rapidly evolving field. The two main research areas are

- System management
- Performance and fault management

System management research is especially concerned with the problem of how to parallelize system management. This issue is important because traditional centralized management approaches, although simple, are not scalable and may not allow effective management or intervention when the network most needs system management [12]. The coordination of distributed system managers is being actively investigated. Several ISO standards, Association Control Service (ACS), Remote Operation Service (ROS), and Open Distributed Processing (ODP) have been, or are being, produced to indicate a framework that can be adapted to achieve distributed computing (or cooperation) within the OSI network management framework. Despite this standardization effort, applied research efforts and distributed system management applications with a few exceptions [16] are being developed today by either extending the much simpler and widely used SNMP or creating ad-hoc distributed management solutions targeted for particular applications [7, 6, 15, 5]. Distributed system management will play an important role in the future because it will be the only alternative to managing ever-increasing, complex heterogeneous telecommunication and distributed computing systems.

Performance and fault management is a system management aspect that has received much attention in the past few years because of its complexity. Again, the major difficulty of effectively managing a complex system is given by its heterogeneity. This aspect of system management is further aggravated by the fact that faults are asynchronous events originating from the leaves of an NM hierarchy and very quickly can exhaust the bandwidth reserved for NM. In addition, as mentioned in the introduction, the alarms generated by faults typically generate a great amount of redundant information that is hard to analyze in isolating the root cause of a problem. A great amount of effort has been put into the development of techniques that try to limit the amount of redundant information to be analyzed (filtering) and try to automatically reason about the alarm traffic observed so as to indicate possible root causes. Most work on this particular subject is driven by particular needs and is targeted for specific NM applications. In [8, 3, 10, 14] topological information stored in the correlation software plays a primary role in isolating and identifying the root causes of faults. A detailed description of the network (with varying degrees of abstraction), plus some heuristics, is used to automate the process of isolating a fault.

Not much exists today with respect to modeling of NM systems. Traditionally, NM systems have been built with a bottom-up approach both because they were a complementary addition to existing telecommunication structures and because in the past they have not posed many problems with regard to their complexity and size. The work that relates the most to ours is described in [11]. The authors describe an integrated fault and alarm generation simulator and alarm correlator software that can be used to simulate the behavior of their NM system in the presence of faults. Their simulation and fault diagnostic systems are highly integrated and hold a detailed description of the NM system by instantiating and connecting the set of specific equipment modules forming their network. Faults and alarms are injected into the modules by a fault generator process that decides where and when the faults must appear in the network. Because this work was focused on a particular set of equipment modules, it lacks in generality and thus cannot be applied to the study of NM systems in general. In addition, the alarm-generation function seems to be completely centralized and unrelated to the network characteristics.

3. Our Models

The abstract models we use to conduct performance studies are meant to be very general and independent of their particular utilization. For small NM systems, they could be amenable to analytic studies. However, we found that, because interesting NM systems are big and complex, they are best studied by simulation.

3.1 Network Management Model

We model an NM system as a complex interconnection of NEs. All NEs are connected to a local NM manager to which they report their state. Each local NM manager may also be connected to a higher-level manager that is responsible for summarizing and presenting the status of the system to a human operator. We assume that the managers have dedicated but limited network and processing resources for management traffic and a limited computational speed. Each manager also has a local information database (MIB) in which it keeps up-to-date information about the status of the system it manages. Local managers do not communicate directly, but can indirectly communicate through the higher-level managers with which they have connections. Under normal operation (without unusual conditions) NM managers will send messages to the NEs so as to perform day-to-day NM maintenance tasks. The maintenance tasks include queries about statistical performance figures, queries about topology information, control messages aimed at reconfiguring the system, and any other tasks that can be carried out remotely. In a normal mode of operation (when everything works correctly) messaging is initiated by the NM managers, and the NEs respond to the queries or control messages with acknowledgments.

3.2 Fault Generation Model

In our model each NE has a mean time between failure (MTBF) parameter associated with each of the possible causes of unusual conditions (or faults). The MTBF figures of each NE dictate the likelihood for an NE to incur in a fault. Because the MTBF figures are given on a per-fault basis, it is possible to accurately describe the reliability of an NE with respect to its particular characteristics. Randomly, NEs incur into faulty conditions with a frequency directly derived from their MTBF parameters.

3.3 Alarm Generation

With unusual conditions or errors, the NEs generate a trap. Traps will cause an alarm message to be sent to the NM manager to describe the cause of the unusual condition. Once an NE has sent the alarm message for a particular condition, it will not send any more messages for the same fault until its state is reset by either the NM manager or by the hardware. It is possible, however, that the same NE may send more alarm messages, which describe different and subsequent unusual conditions.

3.4 Fault Propagation Model

As mentioned in the introduction, one of the most challenging aspects of performance estimation of NM systems is that the behavior of an unusual condition (or alarm) generated by a real system is very hard to model with standard analytical techniques [1]. It has been observed that in most cases faults occur in bursts because any change of the behavior of a single node of a complex network can perturb, and therefore may cause faults in, other nodes of the same network. This phenomenon is known as fault correlation and is the cause of undesirable NM traffic. We create a fault propagation model allowing the capture of the complex interrelations of a real network [1] and therefore yielding a more realistic model of its behavior. Once a fault occurs in an NE, it will be propagated, according to a given probability function, to other NEs. The generation of random numbers and user-specified parameters will determine where the fault is propagated.

3.4.1 Fault Interrelations

We model fault propagation by constructing a directed graph holding a description of the various fault interrelations. Each node of the graph is an NE, and each directed edge coming out of a node has associated with it a weight factor (see Fig. 1). When a fault is started in an NE node, the outgoing link weights are the probabilities that the fault propagates to the nodes pointed to by the links. It is important to realize that these links are only logical, not physical, and that they are implanted by the system designer for modeling purposes only. The weights of the links are expressed as a function of the alarm we are trying to model.

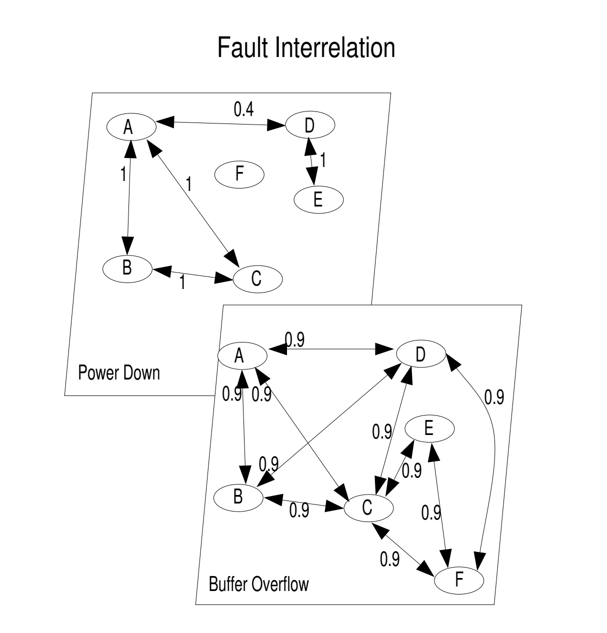

Fig. 1. Alarm correlation for a 6-NE network.The correlations are entered on an alarm basis to capture different characteristics of the system.

Fig. 1 is an example of a simple alarm interrelation model of six NEs with respect to power outage and buffer overflow conditions. All the links are shown to be bidirectional, but this is not required by the model, in which the links can be specified as directed edges.

To better understand the model, it is necessary to relate the coefficients on each link with the physical characteristics of the network that could be modeled by this example. In the case of power outage, we show that NEs A,B,C and D,E are related by a link probability of 1 (always true) This means that we are modeling a system where these NEs share a common power source (if any NE of the groups A,B,C and D,E experiences a power failure, all other NEs in the same group also do). A possible configuration that would be modeled by these probabilities is the one in Fig. 2.

Things can look very different with respect to other conditions. Let us take, for example, the buffer overflow condition to illustrate how some different system characteristics can be modeled by this other alarm.

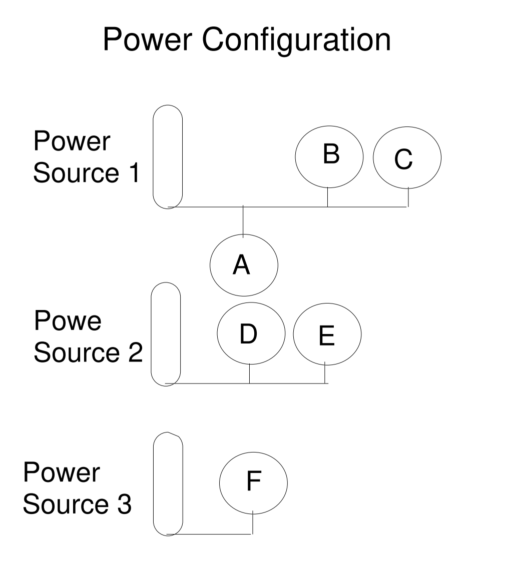

Fig. 2. A possible power supply configuration. NEs A,B,C, NEs D,E and NE F are serviced by three independent power sources. A and D are geographically close.

Fig. 3 shows a possible interconnection of the same six NEs that serve three different users. We assume that the system is heavily loaded and that if a user cannot use an NE because of buffer overflow, it can attempt to use its other available NEs. In this scenario, if one NE of a user gets congested, there is a good chance that the other NEs serving that user will also be flooded with service requests because of the decreased bandwidth availability. Our model can directly reflect this behavior by assigning weights in such a way as to interrelate this kind of condition to the particular usage pattern. As shown in Fig. 1, NEs A,B,C,D, NEs C,E,F and NEs F,D are strongly interrelated with regard to buffer overflow while, for example, an overflow on NE F cannot cause problems to NEs A and B because they are not shared by the same user.

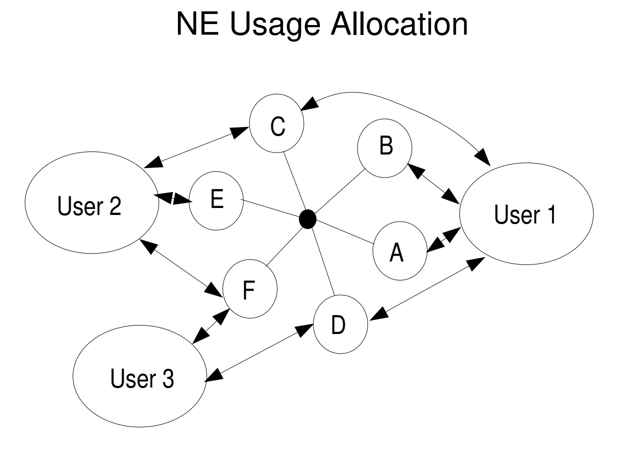

Fig. 3. User 1 uses NEs A,B,C,D only. User 2 uses NEs C,E,F and user 3 uses NEs F,D. If a user cannot access one of its NEs, it can try one of the other available NEs.

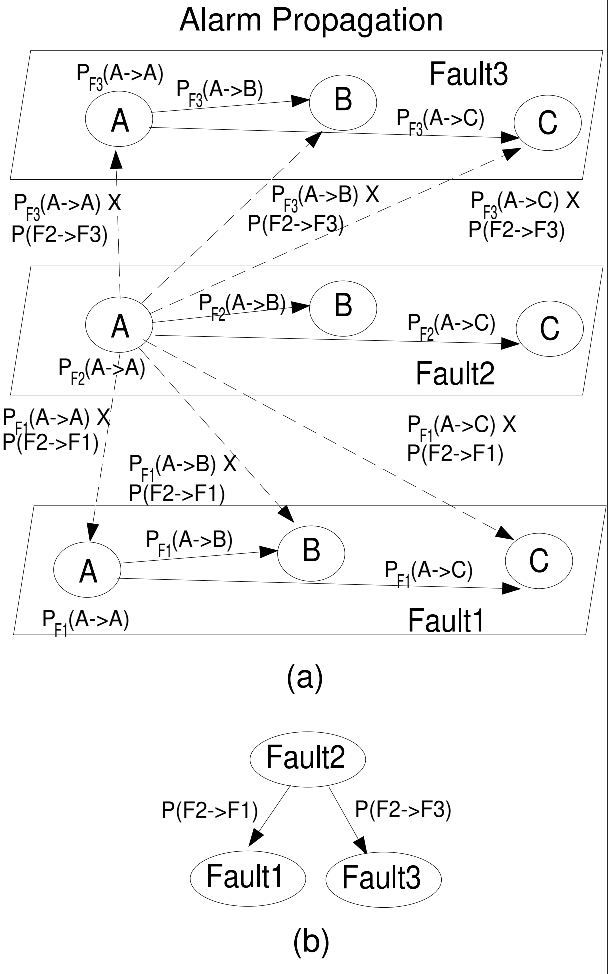

Fig. 4. The fault correlation probabilities and the intra-fault dependency probabilities are combined to interrelate different faults on different NEs.

3.4.2 Hierarchy of Alarms

So far, we have described how the NEs of a network are interrelated with respect to individual faults. This aspect of the model is enough to emulate the behavior of systems in which the different types of alarms are independent from one another or a system with only one type of alarm. Real systems, on the other hand, display a behavior that is more complex. In the typical case, severe alarms cause other related alarms either on the same NE or on other NEs. For example, an NE that experiences a power outage and therefore tries to divert network traffic to other NEs might indirectly cause network congestion and therefore be the cause of other kinds of alarms related to that condition. In the example in Figs. 1, 2, and 3, it is likely that if NEs A,B,C experience a power failure, they will indirectly cause buffer overflow on NEs D,E,F because users 2 and 3 will have decreased bandwidth availability. To model such apparently chaotic behavior it is necessary to add, in the model, some extra information that captures the hierarchical nature of the various conditions. In our experiments, we have used a system-wide interrelation graph that takes into consideration this important aspect of modeling fault behavior.

Fig. 4b is an example of such an interrelation graph, which in this case interrelates faults of types 1, 2 and 3. The direct edges between faults describe the probabilities that each fault has to cause other faults (a nonexisting link is assumed to have 0 probability). By combining the fault interrelation information described earlier with this dependency graph, one is able to calculate the probabilities that a given fault at a given NE will trigger other unusual conditions in the system.

Fig. 4 is a pictorial representation of the effect of combining the two probabilities. In this figure, NE A is in the faulty state "Fault2" and in Fig. 4a shows how this fault can propagate to other faults and other NEs. The horizontal solid links for each alarm (Fault1, Fault2, and Fault3) have weights that determine the likelihood that NE A will cause other NEs to enter each of the faulty states. The diagonal dotted links are not stored in the model representation but are obtained by the multiplication of the horizontal links and the interfault dependency graph weights. The dotted links are associated with the probability that Fault2 on NE A will generate other kinds of faults (Fault1 or Fault3) on the NEs of the system. It is important to note that, for each alarm, NEs have a link that allows them to "infect" themselves with multiple different alarms (in most cases this link has a weight of 1).

4. Simulation Experiments

The complex interactions of a large interconnection system may often cause the occurrence of an initial fault to be the direct or indirect cause of other conditions. An analytical treatment of our probabilistic model that took into account dependency between all the probabilities of a complex network would be limited to relatively simple cases. Simulation, as often happens in these cases, is the only viable alternative to doing detailed performance studies. Simulation allows instantiation of an arbitrarily complex model without forcing the system designer to reduce modeling accuracy. For this reason, we implemented a parameterized simulator [10] for the explicit purpose of studying NM systems under faulty conditions.

The simulator was implemented with Csim [18]. Csim is an extension of the C language that allows easy creation of a process-oriented event-driven simulation engine just by augmenting C procedures with some simple simulation and modeling constructs. We simulate the NM system by mapping each NE or NM manager to a separate process, and we simulate the messaging mechanism with mailboxes. The different processes created through Csim allow simulation of the system of NEs and NM managers as if they were executing in parallel. We carefully model network contention by allowing only one entity at a time to transmit data on a shared transmission line, and we also take into account bandwidth parameters to regulate the flow of information. For the moment, we ignore protocol stack overhead and lump all transmission delays into latency and throughput parameters specified by the user for each physical link of the network. In addition to transmission delays, we model service time at each node of the network. This parameter simulates the interval from the time an NE experiences a fault to the time it starts sending an alarm message and, and more important, the time it takes an NM manager to process each alarm message.

We developed some analytical models of simple NM systems when they experience faults, and we compare them to results we obtained with simulation. The development of the analytical models will give an account of how difficult it is to symbolically evaluate complex interconnection systems, and at the same time (for some simple cases) will allow us to compare theoretical results with the results obtained with our simulator. In addition, we will compare different theoretical models to show how results are affected by the degree of accuracy of these models.

For all experiments we will equate NEs MTBF to the inverse of the arrival rate λ and the NM manager processing time to the inverse of the service rate μ. For these experiments, we simplify our model by assuming that each NE is fixed "immediately" after it generates an alarm, transmission channels have zero delay, and the system has infinite queues. We also restrict our analysis to the first two levels of the NM hierarchy.

In all experiments, we measure the average number of messages, in the system, that have not been serviced yet because this metric is easy to derive from both analytical models and our simulator.

4.1 Exponential Service Time with Constant Batch Size

In this experiment we model a system consisting of multiple NE. A fault occurring in an NE automatically causes a `batch' of b alarms. The interarrival time between the batches is exponentially distributed as in the previous experiment, but at each occurrence of a fault a fixed number of service requests is issued. This behavior approximates what would happen in an NM system when a fault triggers other related (downstream) faults. Although this model is not very useful in our case (one could argue that if b alarms are always triggered in parallel, they could be grouped as a single alarm and be processed in parallel, thus reducing the system to an M/M/1 model), we compare a constant batch-arrival system with both an equivalent M/M/1 system and our simulation to gain more insight into the relative accuracies of these models and to give further confidence in the simulator output. In this experiment, we fix the batch size to 5 and vary $\frac{\lambda}{\mu}$.

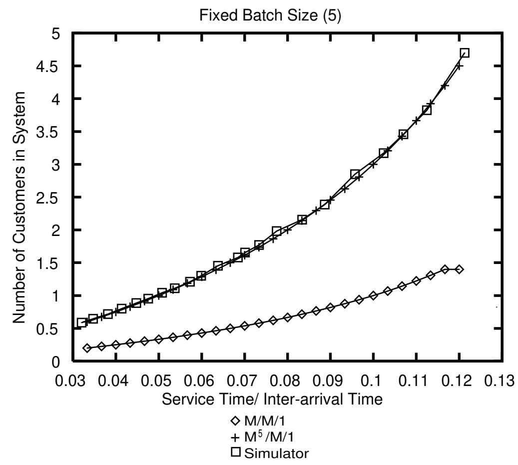

Fig. 5. Utilization ratio of a system with exponential arrival rate, exponential service time, and batch size 5.

For M/M/1, the number of customers in the system is adjusted to

$\frac{b\lambda}{\mu - b\lambda}$, while for the fixed batch size model we have

$\frac{\rho(1+b)}{2(1-\rho)}$.

From Fig. 5 we notice that while the simulator closely emulates the batch-size model, the M/M/1 system is not adequate to model this particular behavior.

4.2 Exponential Service Time with Variable Batch Size

In this more realistic experiment, we allow all NEs of the NM system to be part of each fault with a probability of 0.5. In other words, we generate faults with an exponential distribution, and at each fault occurrence all NEs in the system have a 0.5 probability of generating an alarm for that fault. This method can model the behavior of an NM system in which the number of generated alarms depends on the particular state of the system at the time of the fault.

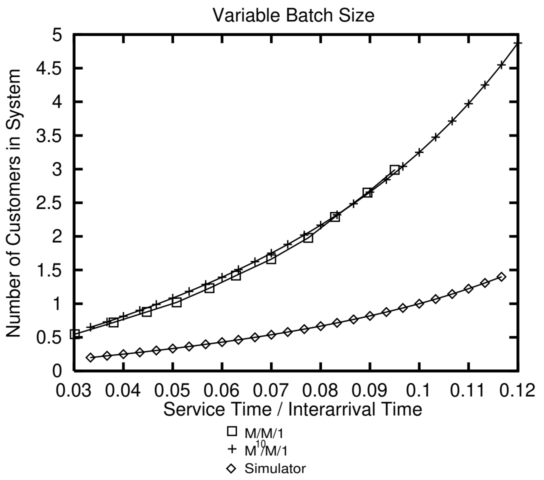

Fig. 6. Utilization ratio of a system with exponential arrival rate, exponential service time, and variable batch size.

This experiment gives some insight on how complex an analytical study can become as soon as the behavioral model is made slightly closer to reality (we are still very far from a real behavior because of a lot of the assumptions).

The number of customers in a single-server system with exponential arrival rate, exponential service time, and a batch size probability function g is given in terms of its z-transforms as $P(z) = \frac{\mu P0(1-z)}{\mu(1-z)-\lambda z[1-G(z)]}$, where G(z) is the z-transform of the distribution of the bulk size.

The probability that, out of N total NEs, n NEs will generate an alarm (assuming that all the NEs have the same probability of generating an alarm once a fault occurs) is given by the binomial distribution $\binom{n}{N}P(1-p)^{(N-n)}$, where p is the probability that the NEs will generate an alarm.

From [4] we derive the average number of customers in the system as being $\rho + \frac{\rho^2}{1-\rho} + \frac{\rho(\sigma^2+\overline{a}^2-\overline{a})}{2\overline{a}(1-\rho)}$ where $\overline{a} = G^{'}(1) = np$, $\rho = \overline{a}\frac{\lambda}{\mu}$ and $\sigma^2 = np(1-p)$.

We compare this theoretical value with both an adjusted M/M/1 system and the simulator results for p=0.5 and n=10, and we see again that the M/M/1 model is not adequate and that the simulator matches the analytical model for this system. Fig. 6. shows the average utilizations for the different experiments as a function of $\frac{\lambda}{\mu}$.

5. Conclusion

We have described a new model for the description and simulation of fault reporting in NM systems. Our model is very general and allows us to describe any NM network by appropriately choosing the model parameters. This model will allow advances in NM system design and performance evaluation by offering the possibility of exploring, through simulation, different design alternatives in a flexible synthetic environment. Because the researcher can generate realistic traces of a faulty NM system without having to experiment directly in the field, a much more efficient methodology is used for testing NM design ideas.

We plan to apply our modeling ideas to existing telecommunication networks and to study their behavior through simulation. Apart from validating our models by comparing our performance results with real network data, we will be able to perform scalability studies to determine potential system-level performance bottlenecks that could appear in future expansions of the network. By enabling NM performance evaluation at the design stage, we will be able to detect and analyze critical aspects of NM system behavior before a system is built, thus preventing an enormous waste of resources.

References

[1] S. Aidarous and T. Plevyak. *Telecommunications Network Management into the 21st Century*. IEEE Press, 1994.

[2] A. Bouloutas, A. Calo, and A. Finkel. Alarm correlation and fault identification schemes in communication networks. Technical Report RC 17967, IBM, 1992.

[3] L. Chang, W. Chan, J. Chang, P. Ting, and M. Netrakanti. A network status monitoring system. In *GLOBECOM 89*, volume I, 1989.

[4] M. L. Chaudhry. *A First Course in Bulk Queues*. New York: Wiley, 1983.

[5] T. P. Collins. Polycenter license system: Enabling electronic license distribution and management. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 108. Chapman & Hall, 1995.

[6] S. Crane, N. Dulay, H. Fossa, J. Kramer, J. Magee, M. Sloman, and K. Twidle. Configuration management for distributed software services. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 29. Chapman & Hall, 1995.

[7] J.C. Gregoire. Models and support mechanisms for distributed management. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 17. Chapman & Hall, 1995.

[8] K. Houck, S. Calo, and A. Finkel. Towards a practical alarm correlation. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, pages 226–237. Chapman & Hall, 1995.

[9] I. Katzela, A.T. Bouloutas, and S.B. Calo. Centralized vs. distributed fault localization. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, pages 250–261. Chapman & Hall, 1995.

[10] R. Manione and F. Montanari. Validation and extension of fault management applications through environment simulation. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, pages 238–249. Chapman & Hall, 1995.

[11] R. Manione and F. Montanari. Validation and extension of fault management applications through environment simulation. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, pages 238–249. Chapman & Hall, 1995.

[12] K. Meyer, M. Erlinger, C. Sunshine, G. Goldszmidt, and Y. Yemini. Decentralized control and intelligence in network management. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 4. Chapman & Hall, 1995.

[13] M. Moller, S. Tretter, and B. Fink. Intelligent filtering in network management systems. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 304. Chapman & Hall, 1995.

[14] Y. A. Nygate. Event correlation using rule and object based techniques. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 278. Chapman & Hall, 1995.

[15] A. R. Pell, K. Eshghi, J-J. Moreau, and S. J. Towers. Managing in a distributed world. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 94. Chapman & Hall, 1995.

[16] A. W. Pratten, J. W. Hong, M. A. Bauer, J. M. Bennet, and H. Lutfiyya. A resource management system based on the odp trader concepts and x.500. In A. S. Sethi, Y. Raynaud, and F. Faure-Vincent, editors, *Integrated Network Management IV*, page 118. Chapman & Hall, 1995.

[17] M. Reise. Model based diagnosis of networks: Problem characterization and survey. In *OEGAI-91 Workshop on Model Based Reasoning*, 1991.

[18] H. Schwetman. Csim: A c-based, process-oriented simulation language. Technical report, MCC, 1989.

[19] N. Shroff and M. Schwartz. Fault detection/identification in linear lightwave. Technical Report CU/CTR/TR 243-91-24, CTR, 1989.

[20] C. Wang and M. Schwartz. Identification of faulty links in dynamic-routed networks. *IEEE JSAC*, 11, 1993.