Computer Network Reliability Dynamics Modeling: An Automatic Service Stabilization

Benson Moyo[1], **and Ndabezinhle Soganile**[2]
[1]Computer Science and Information Systems Department, University of Venda, Thohoyandou, Limpopo, South Africa; Email: benson.moyo@univen.ac.za
[2]Computer Science and Information Systems Department, University of Venda, Thohoyandou, Limpopo, South Africa; Email: ndabezinhle.soganile@univen.ac.za

Abstract - *Dependence on computer networks is now a reality. The networks are continuously becoming larger, heterogeneous and more complex in size and functionality. Any computer network service relies on several component interactions. This might involve mutually communicating hardware and software. The main task is to guarantee network service in the presence of network faults. In this work we present a design of a model to identify, predict, evaluate and neutralize faults on a computer network. The approach incorporates the capability of Dynamic Bayesian Networks to diagnose, predict and forecast faults and evaluate the magnitude of the network service degradation. The model is complex enough to diagnose and yet simple enough to avoid time and space complexity on the network. Fault thresholds to satisfy probability of error occurrence are established. We present model simulated results to demonstrate the applicability of proposed model.*

Keywords: Network reliability, fault, Dynamic Bayesian Network

1 Introduction

Dependence on computer networks has become a necessity to many organizations both profit making and nonprofit making including individuals. There is a whole explosion of services based on the network platform. The computer networks are themselves becoming larger, complex, ubiquitous, flexible and dynamic. Devices may be added, undergo repairs, be upgraded, and be removed from the network at any given time.

In order to access the correct service, a user needs several components to interact at least according to their requirement specification to perform one or more specified operational function. This might involve mutually communicating applications software, system software, protocols and communicating hardware and transmission media. The final output from the network is the sum total of the interaction of all the necessary components and subcomponents and protocols. Each component, each interface, each protocol plays an important role in the overall outcome. How can we guarantee high degree of network service survivability in the presence of network faults, malfunctions, malware attacks and software design flaws? The user is not interested in whether there has been a removal or repair of a device or debugging or upgrade of a protocol but is concerned with the availability of a an expected service at a given point in time as dictated by requirements and needs. Therefore organizations need some solid form of trust from the network. The confidence is brought by the application of sound reliability models.

In this work we present a design of a model to identify, predict, evaluate and neutralize faults on a computer network. In section 2 we present a brief overview of computer network faults classes. The Dynamic Bayesian Network approach to diagnose, predict and forecast faults and evaluate the magnitude of the network service degradation is also presented in section 2. We present model simulation results in section 3 and draw our conclusions in the subsequent section which is section 4.

2 Methodology

The body of work includes in depth analysis of the literature and work that has been done in the field of fault management. The relevant literature review guides the designing of computer network hybrid reliability model. Both secondary and primary sources of information are used to enhance the researchers' knowledge in the field of study and enable them to design a model based on the effort done by other researchers to the same end. Probabilistic techniques, Dynamic Bayesian Networks, and simulation methods are used as the foundation and the building blocks of this work. Network faults are identified using the Network Management error codes, and the Simple Network Management Protocol trap messages.

The simulation is designed and implemented using C++. The data used is collected from management agents via Simple Management Protocol messages (SNMP traps) and network system log-files. For simulation purposes the random occurrence of faults, the minimal standard random number generator as recommended in [1] is used. The probability outcomes are then compared with the statistics from the collected data as experimental control.

2.1 Overview of computer network faults

A network fault is defined as an abnormal condition or defect at the network component, equipment, or sub-system level which may lead to an error which may in turn lead to a failure [2]. Network faults constitute a class of network events that can cause other events but are not themselves caused by other events as explained in [2].

Therefore a network failure is a result of a network system state error and the generator of an error is a fault. An error can propagate through a network and cause hardware and software failure to otherwise faultless hardware and software subsystems. A failure is the manifestation of the error that is observable by the client. The client might be a human being or another network component or system [2].

Systems have faults but as long as that fault is not activated and it has not triggered an error, we cannot talk of a failure. Failure of a computer network is with respect to the abnormality in service provision visible to a client.

Computer network faults can be classified as permanent, transient and intermittent [2]. This is the taxonomy of faults based on their temporary effects. Taking an assumption that the active period of a fault is the interval during which the fault has a negative influence on the network, and benign period is the interval where the fault is not influential. Therefore a formalized classification definition is as follows: The permanent fault has an infinite active period. It reflects irreversible physical changes to a system or subsystem. Transient faults have a finite active time interval followed by an infinite benign period. These are generated by temporary network conditions like loss of signal in wireless network or an active attack by a hacker. Intermittent faults have finite active period and finite benign period, in other words these are recurring. Intermittent faults are internal in nature like network congestion.

Classification of faults based on the behavior are presented in [3], these can be summarized as crash faults (component either completely stops operating or never returns to a valid state); omission faults (component completely fails to perform its service); timing faults (component does not complete its service on time or suffers from synchronization) and Byzantine faults (faults of an arbitrary nature).

Although errors induced by transient and intermittent faults manifest very similarly, two main criteria as observed in [4] may be used to determine the type of the fault that generated the error. Firstly, failures generated by errors induced by intermittent faults tend to occur as a cluster at the same location, when the fault is activated. Secondly, replacement of the culprit component or subsystem removes the intermittent fault; by contrast failures generated by errors induced by transient faults cannot be eliminated by repair. The cause of a transient fault cannot be traced to a defect in a well identifiable part of the network. For simulation purposes we use some Simple Network Management Protocol Management Information Base (NMP MIB) data and network system logs to establish an estimate of transient fault rate and intermittent fault occurrence patterns. However this is highly contextual as each situation is dependent on a myriad of network characteristics from hardware brands and type to software and parameter configuration. A permanent fault induces a permanent error which in turn induces a permanent failure irrespective of the context. On the other hand a transient fault induces a transient error which subsequently induces a transient failure. Lastly an intermittent fault likewise induces an intermittent error which also triggers an intermittent failure.

According to Mouhammd [5], faults can be split into two categories, soft and hard faults. Wear or damage constitutes hard faults. These may be either hardware or software. The soft fault stem from poor engineering principles such as incomplete design. We focus on the soft faults in this study though there is overlapping.

Table 1: Fault classes

Faults Classes		
Permanent	Transient	Intermittent
Hardware clash Power outages Software clash	Malware Software bugs Hackers Buffer overflows	Packet loss Congestion End-to-end delay

The objective of understanding computer network faults is to be able to identify them according to their classes and as such it facilitates the modeling process.

Network reliability is the probability that a network will perform satisfactorily for at least a given period of time when used under stated conditions. It is an important attribute of a computer network as a system. Network reliability mostly deals with long term, or average behavior of the network. It is a property of the network and evaluates characteristics such as failure rate and failure density, architectural properties. The identification of faults is an important step towards reliability modeling of a system. In figure 1 below we show the relationship between failures, errors, and faults. A fault may develop into an error which may develop into a failure. However not all errors can develop into a failure. Some errors may only develop into failures only under certain conditions. The Millennium Bug, in Year 2000 is an example of a failure that was triggered by time.

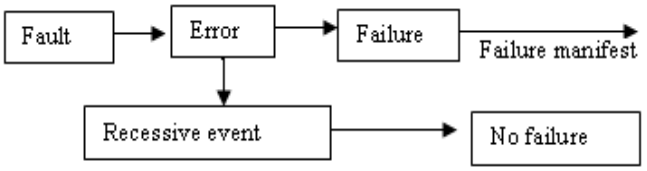

Figure 1: Fault, error failure relationship

A computer network is a repairable system. This means that if a component fails we can either replace the subcomponents that failed; this is referred to as repair maintenance, or replace the whole component by a new one

and this is referred to as preventive maintenance. All this is done to recover from a failure that would have been caused by a fault. In order to design a reliable system we need to specify the faults that the system may be subjected to. Therefore fault assumption is important in the whole system reliability modeling process. We have identified and classified the faults. Now we present methods for dealing with them.

2.2 Bayesian belief network

Bayesian networks provide a complete description of a given problem domain based on causal relationships. The cause-effect relationships enable both forward and backward reasoning. Every entry in the full joint probability distribution can be calculated from the information in the network. Bayesian networks represent full probability models in a compact and intuitive way. These networks can address problems in diagnosis, prediction, forecasting, information retrieval, knowledge representation and many other domains [6].

As well as being a complete and nonredundant representation of the domain, Bayesian networks are more compact than the full joint distribution and their time and space complexity is lower than that of other models like the Markov Chain Models. This property is what makes it feasible to handle domains with many variables. Bayesian networks are sparse systems [6]. This means that it satisfies the property of being locally structured. The premise for locally structured systems is that, each subcomponent interacts directly with only a bounded number of other components; regardless of the total number of components this will imply less complexity exponential explosion.

In the Bayesian Network framework the independence structure in a joint distribution characterized by a directed acyclic graph, with nodes representing random variables (which can be discrete or continuous, and may or may not be observable), and directed arcs representing causal or influential relationship between variables. The conditional independence assertions about the variables, represented by the lack of arcs, reduce significantly the complexity of inference and allow the underlying joint probability distribution to be decomposed as a product of local conditional probability distributions (CPD) associated with each node and its respective parents. The semantics of Bayesian networks can be viewed in two ways that is as networks that represent joint probability distribution or as an encoding of a collection of conditional independence statements. The two views serve to guide the construction of the network and the designing of inference procedures. [6].

2.2.1 The rationale behind employing Bayesian network for this study

As aforesaid Bayesian Networks, are space efficient data structures for encoding all of the information in the full joint probability distribution for the set of random variables that characterize a problem space. It uses the fact that in real-world problem domains, the dependencies between the variables are generally local. The Bayesian network allows one to compute any value in the full joint probability distribution of the set of random variables. It represents all of the direct causal relationships between variables which are an advantage in the determination of failure causes. It can be used to reason forward (top-down) from causes to effects (predictive reasoning) or backward bottom-up) from effects to causes (diagnostic reasoning). In other words we can infer faults sources from errors and vice versa.

2.2.2 Constructing Bayesian network for a network fault

Let X represents faults of arbitrary nature. Let the Bayesian Network for the set of fault variables $X=\{x_1, x_2, \ldots x_n\}$ represent a joint probability distribution: $P(x_1, x_2, \ldots x_n)$. Then we writing the joint probability as a conditional probability using the chain rule we have:

$$P(x_1, x_2, \ldots x_n) = P(x_n|x_{n-1}\ldots x_1)P(x_{n-1}\ldots x_1) \quad (1)$$

Then reducing equation 1 to each conjunctive probability to a conditional probability we have:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i | x_{i-1,\ldots.} x_1) \quad (2)$$

The expression is referred to as the chain rule. Therefore for every fault variable X_i in the network provided that Fault-Evidence $(X_i) \subseteq \{X_{i-1}\ldots X_1\}$, we can express its probability as:

$$P(X_i|X_{i-1}\ldots X_1)=P(X_i|\text{Fault-Evidence}(X_i)) \quad (3)$$

Let X_t, denotes a set of unobserved fault state in time t, and e_t denotes the observed error. The observed error implies fault evidence. Therefore to predict the future fault state of X we have to extend equation 3 to obtain the following expression:

$$P(X_{t+k}|e_{1:t}) \text{ for some } k > 0. \quad (4)$$

For example, the expression might mean computing the probability of network buffer overflow fault five time units from now given all the observations of network buffer overflow errors or failures up to now.

It is not only forward reasoning that is of interest but also retrospective reasoning. The diagnostic reasoning in helps establish fault sources. This fault hindsight entails computing the posterior distribution over a past fault state, given all errors or failures to the present point in time. That is, from equation 4, we wish to compute $P(X_k|e_{1:t})$ for some k time units such that $0 \leq k < t$. For instance if we want to compute the probability that there was network congestion five units of time ago given the network congestion based errors or failures up to now. In this case k=5 units of time.

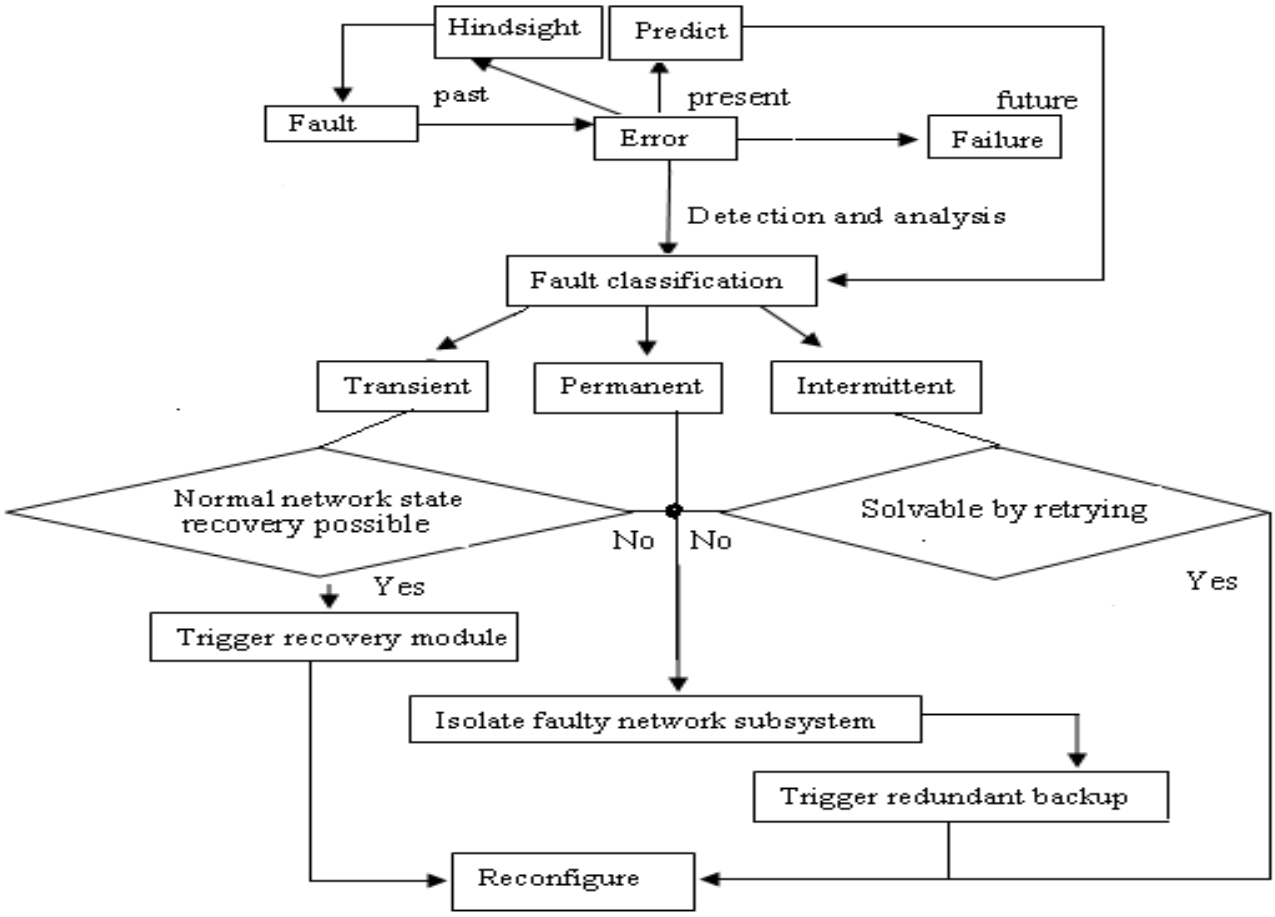

Figure 2: Proposed Reliability architecture

3 Results and discussion

Using the information obtained from a single organizations' network over a period of six months from SNMP, and network log files we obtained the fault rates shown in table 2. If there is a fault the probability that it might be associated with a protocol either poorly configured or corrupted is 0.48.

Table 2: Basic component fault rates from data collected

Basic network component	Fault rate
Switch	0.100
Bridge	0.130
Server	0.260
Network adapter	0.004
Router	0.210
Protocol	0.480
Transmission media	0.340

Let IF_{t-1} represent intermittent fault probability state variable in time t-1, and we also assume that:

$P(IF_t|IF_{t-1}=true)=0.013$, $P(IF_t|IF_{t-1}=false)= 0.987$ and $P(IE_t|TF_t=true)=0.46$, $P(IE_t|IF_t=false)= 0.29$. IE_t represents the intermittent error in the network. We predict for the next nine time units (correct to 9 decimal places) and the results are shown in table 3 and figure 3. The results demonstrate the temporary nature of this type of fault. In some cases only a single run will result in the belief that a fault does not exist as proved by our simulation model. However due to its recurrence nature also we found that it can approach absolute values in some probabilistic instances.

Table 3: Intermittent fault probability over time

Time units into the future	Intermittent fault probability changes with time
1	0.110400000
2	0.006791759
3	0.001101247
4	0.000785880
5	0.000768500
6	0.000767543
7	0.000767490
8	0.000767487
9	0.000767487

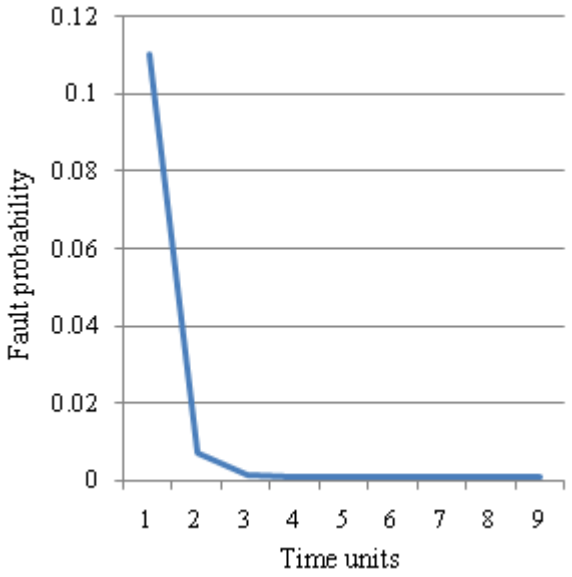

Figure 3: Intermittent fault probability trajectory over time

Given the initial belief that the probability of a transient fault is 0.5, TF_{t-1} represent transient fault probability state variable in a time t-1, and we also assume that $P(TF_t|TF_{t-1} = true) = 0.85$, $P(TF_t|TF_{t-1} = false) = 0.15$ and $P(TE_t|TF_t = true) = 0.49$, $P(TE_t|TF_t = false) = 0.027$. TE_t represents the transient error in the network. We predict for the next ten time units and the results are shown in figure 4. These results demonstrate the persistence of this type of fault and indicate a probability increase on the significant disruption of a particular network service. Figure 4 below shows the graphical representation of the results, this type of fault needs to be eliminated to restore normal service.

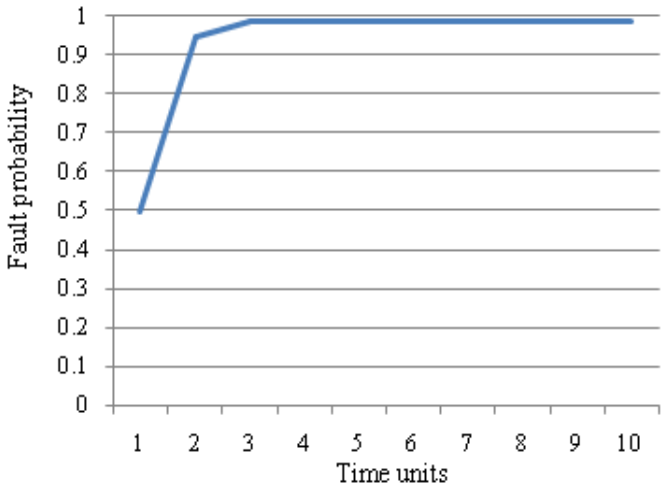

Figure 4: Transient fault probability progression over time

Table 4 was generated by our model simulation. However there is varying degrees of uncertainty when it comes to calculating the probability of a fault and classifying it as a condition necessary and sufficient for a failure occurrence.

Table 4: Forward inference on fault classification

Fault	Error	Failure type	Automated decision
Router interface out of synchronization	Packet loss	permanent	Temporarily isolate router by switching to standby router.
Timing mismatch	Time out	Intermittent, permanent or transient	Retry, or activate standby
Power cut	Blackout	Permanent	Switch to UPS
Broadcasting burst packets	Congestion , Bandwidth saturation	Intermittent	Retry
Network Protocol misconfigurations	Invalid routes	Transient	Switch to alternative node and reconfigure

4 Conclusions

In this paper we have considered the applicability of Bayesian networks for reliability modeling. Faults, errors and failures are probabilistic in nature therefore Bayesian Networks constitute a sound probabilistic rigorous mathematical modeling framework to tackle this domain. Network reliability is a research domain that will continue attracting research efforts as networks continue to grow in size functionality and complexity. Each component of a system can be engineered taking care of reliability; however when the systems are put together as subsystems to interoperate, challenges may immerge. Network configurations and reconfigurations, attacks, upgrades all introduce some level of network service uncertainty which is worthy researching. We have surveyed fault taxonomy in order to understand the characteristics and the dynamics of faults. The understanding will help architects to design networks that a more resilient to faults thereby guaranteeing service. The approach taken is based on the assumption that faults exists in any network what is important is how to handle them. We define network faults as a class of network events that can cause other events but are themselves not caused by other events.

The proposed model is designed to identify, localize, categorize the faults and select as well as initiate a recovery process. Prediction is achieved in order to preplan for the future abnormal behavior of the network. A level of redundancy by replicating some critical devices and network links to facilitate network restoration after localization and isolation of a component affected by a permanent fault is proposed.

In furthering this work it is recommended to research on fault parameter estimation and failure distributions and localization without depending on alarms from network management systems.

5 References

[1] J. Banks, B. L. Nelson, and D. M. Nicol, "Discrete-Event Simulation", Pearson Publishers, New Jersey, 5th Edition, 2010.

[2] M. Steinder, and A.. S. Sethi., "A Survey of fault localization techniques in computer networks", Journal of Science of Computer Programming , Vol No. 53, pp. 165-194, 2004, [online] Available from: http://citeseerx.ist.psu.edu/viewdoc, [accessed 20 February 2010]

[3] B. Selic, "Fault tolerance techniques for distributed systems" IBM, Software Group, 2004 [online], Available from: http://www.ibm.com/developerworks, [accessed 24 February 2009]

[4] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods", ACM Computing. Survey Vol. No. 42, Issue No. 3, Article 10 (March 2010), [online], Available from: http://delivery.acm.org/10.1145/1680000/1670680/a10-salfner.pdf , accessed 23 May 2010

[5] Mouhammd Al-Kasassbeh, and Mo Adda, "Analysis of mobile agents in network fault management", Journal of Network and Computer Applications Vol. No. 31, Issue No. 4. pp. 699-711, 2008

[6] S. J. Russell. and P. Norvig, "Artificial Intelligence: A Modern Approach", 2rd Edition, Prentice Hall , New Jersey, 2010

[7] H. Boudali, and J. B. Dugan, "A discrete-time Bayesian network reliability modeling and analysis framework", Journal of Reliability Engineering and Safety, Vol. No. 87, pp. 337-349, 2005 [online], Available from: http://www.sciencedirect.com, [accessed 12 July 2009]

[8] X. Jia, J. Cao, and W. Jia, "A classification of multicast mechanisms: implementations and Applications", The Journal of Systems and Software, Vol. No. 45, pp. 99-112, 1999.

[9] H. Li, and J. S. Baras, "Intelligent Distributed Fault and Performance Management for Communication Networks", Tech. Rep. CSHCN PhD 2002-2, Center for Satellite and Hybrid Communication Networks, University of Maryland, 2002. [online], Available from: http://www.isr.umd.edu, [accessed 12 August 2009]

[10] R. Maxion, "A case study of Ethernet anomalies in a distributed computing environment", IEEE Transactions on Reliability, Vol. No. 39, Issue No 4, pp. 433-443, 1990, [online] , Available from: http://www.ieeexplore.ieee.org [accessed 10 September 2009]

[11] G.R. Weckman, L.R. Shell, and J.H. Marvel, "Modeling the reliability of repairable systems in the aviation industry", Journal of Computer and Industrial Engineering Vol. No. 40, pp 51-63, 2001, [online], Available from: http://www.portal.acm.org, [accessed 5 June 2009]