14

Automation of Cellular Network Faults

Okuthe P. Kogeda and Johnson I. Agbinya[1]
Computer Science Department, University of Fort Hare, Alice 5700, South Africa.
+Center for Real-time Information Network(CRIN), Faculty of Engineering, University of Technology, Sydney, NSW 2007, Australia

1. Introduction

The internet explosion and increasing number of services on offer and subscribers has placed a lot of pressure on cellular network service providers. Cellular network subscribers have different requirements and needs. This requires that the operation of the network be optimal at all times, to attract and retain subscribers. This can happen with proper operation and maintenance of the network itself. The automation of cellular network faults, where these faults are reported before they occur is the approach for avoiding the catastrophic failures that may cause network blackout.

An application of Mobile Intelligent Agents (MIA) in monitoring the network elements for any potential failure of these core objects of the network to be avoided is explored in this chapter. The main concern is the prediction of possible cellular network faults using scenarios extracted from correlation of certain cellular network parameters that may not be evident to human operators. These could be solved using an advanced automated solution. This chapter proposes and discusses the development of a MIA system for computer-aided analysis, simulation and diagnosis based on mobile intelligent software agents (Wooldridge & Jennings, 1995). We propose a framework that utilizes different Artificial Intelligent (AI) techniques and probabilistic methods. Neural networks, fuzzy logic, genetic algorithms, among others, are some of the established artificial intelligent techniques used into software agents (Thottan & Ji, 1999).

In this work we combine a Bayesian Network Model (BNM) with mobile intelligent agents for automating fault prediction in cellular network service providers, in a project called Modelling of Reliable Service-Based Operations Support System (MORSBOSS). The major advantage of using Bayesian network model is that the cellular network faults can be automatically detected based on a similar fault occurrence that the system has experienced previously. The information about the previous fault occurrences can be stored and retrieved from a database. This information shows the causal relation between network elements, network faults and services. It also shows the belief or likelihood of a fault at a particular network element. Fault prediction is therefore based on the historical memory of the system about known faults.

This Chapter is organized as follows: In Section 2, we give a detailed overview of Cellular network faults. Definition, characteristics, causes and classification of cellular network faults are provided in this Section. Methods and algorithms of cellular network faults modeling are provided in this Section. Bayesian network, cellular network modeling process and assumptions are also provided in this Section. In Section 3, we provide Mobile Intelligent Agent (MIA) and reasons for choosing MIA. In Section 4, we present Mobile Intelligent Agent model for cellular network faults prediction. We give reasons for cellular network faults prediction and provide such models in this Section. In Section 5, we present the implementation architecture, stating reasons for the choice of JADE architecture. The experimental results are provided in Section 6 and then we draw conclusion in Section 7.

2. Cellular network faults

2.1 Definition

Cellular network fault can be defined as an abnormal operation or defect at the component, equipment, or sub-system level, which significantly degrades performance of an active entity in the network or disrupts communication. All errors are not faults as protocols can mostly handle them. Generally faults may be indicated by an abnormally high error rate.

A fault can be defined as an inability of an item to perform a required *function* (a set of processes defined for purpose of achieving a specified objective), excluding that inability due to preventive maintenance, lack of external resources, or planned actions (ETSI Guide, 2001).

2.2 Characteristics of cellular network faults

There is lack of a generally accepted definition of what constitutes a behaviour of a normal cellular network fault (Hajji et al., 2001; Hajji & Far, 2001; Lin & Druzdzel, 1997). Therefore, it is very difficult to characterize the cellular network faults accurately. However, there are estimations (based on statistics of the network traffic) as to what characterize a cellular network fault. Cellular network faults are characterized by transient performance degradation, high error rates, loss of service provision to the customers (i.e., loss of signal, loss of connection, etc), poor quality of service provision, delay in delivery of services and getting connectivity, etc.

2.3 Causes of cellular network faults

The main causes of network faults differ from network to network. Managing complex hardware and software systems has always been a difficult task. The Internet and the proliferation of web-based services have increased the importance of this task, while aggravating the problem (faults) in at least four ways (Meira, 1997; Thottan & Ji, 1998; Hood & Ji, 1997a; Lazar et al., 1992):

- The speed of software development and release means less reliable and more frequently updated software.
- Multi-tier and distributed software architectures increase the complexity of the cellular network environment and obscure causes of both functional and performance problems.
- Internet style service construction implies more dynamic dependencies among the distributed software elements of the overall services making it difficult to construct and maintain accurate system models.
- Internet scale deployments increase the number of service elements under a particular administrator's responsibility.
- Many heterogeneous networks
- New innovations means interoperation of different networks must be kept to some level leading to faults.
- Overloading of power supply gadgets, natural disasters, etc.

2.4 Classification of cellular network faults

In addition to the definitions given in Section 2.1, a fault is regarded as an abnormal and/or an accidental condition that is either caused by a defective network element, problem in the network layer or at sub-system level. Such problems often cause a previously functional unit to fail. A cellular network fault can also be viewed as a defect that causes a reproducible or catastrophic malfunction. A reproducible malfunction is one that occurs consistently under the same circumstances.

Cellular network faults may also cause malfunctions and outages. Malfunctions are mostly experienced when the software and hardware are working with some errors. However, outages are often manifested when the software and hardware are completely knocked out; they will not be working at all. When this occurs, ESPs will not only lose revenues but also the customers may shun away. However, in case of outages, ESPs may device contingent plans that are meant to improve services by ensuring that the duration of each outage is kept to minimum time possible.

Cellular network faults are classified as *malfunctions* and *outages*. The model gives an overview of cellular network faults types that are commonly experienced by a cellular network service provider under study. Fig.1 depicts the classification of cellular network faults.

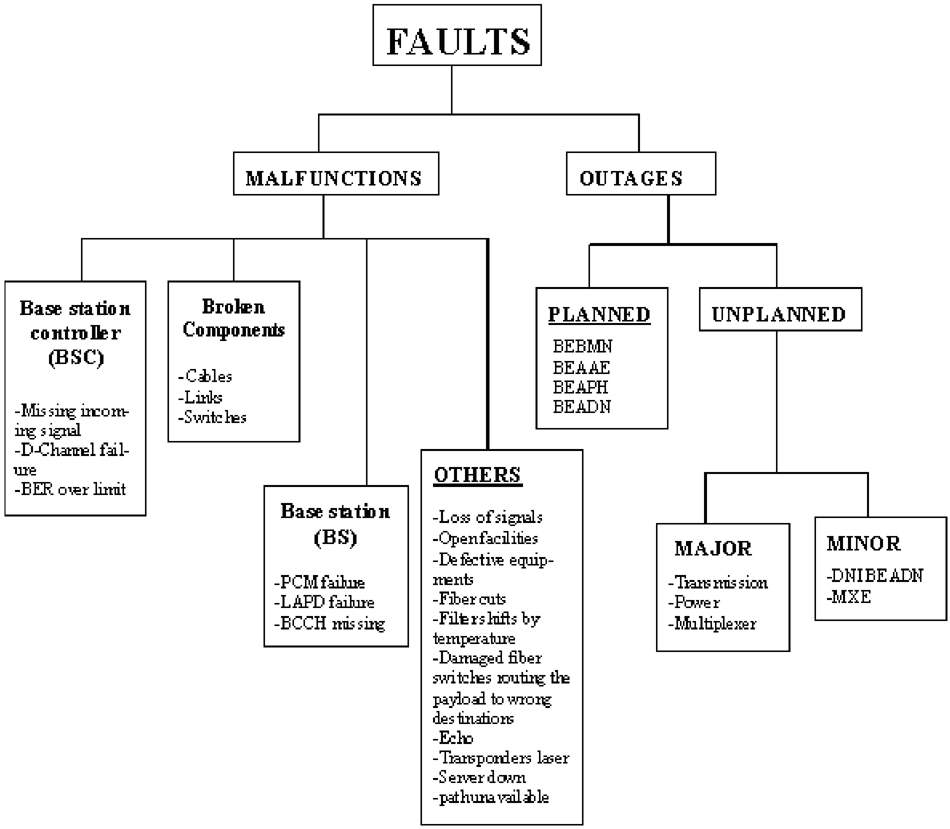


Fig. 1. Classes of Cellular network faults

2.5 Methods and algorithms for cellular network faults modelling

A detailed review of methods and algorithms for faults modeling is provided by Meira (Meira, 1997) and Holst (Holst, 1997) in their respective works. While Meira (Meira, 1997) gives the methods and algorithms for alarm correlation, a summary of some of these methods under machine learning are provided by Holst (Holst, 1997). Most of these methods and algorithms can be utilized in the faults prediction process. Probabilistic approaches and the approaches in which the network entities are modeled as finite state machines are identified (Meira, 1997) for faults identification, detection and prediction in cellular network service providers. Others apply principles defined in non-conventional logics and others adopt *ad hoc* methods to deal with faults modeling.

Among the methods and algorithms include: Fuzzy Logic (Zadeh, 1962, 1965), Artificial Neural Networks (Duda et al., 2001; Widrow, 1989), Decision tress (Mitchel, 1997; Winston, 1992), Model Based Reasoning (Khardon & Roth, 1997; Khardon et al., 1999), Case Based Reasoning (Watson, 1997; Agnar & Plaza, 1994; Klaus-Dieter et al., 1999) Rule Based Systems (Langley & Simon, 1995), Blackboard (Corkill, 1991) and Bayesian networks (Heckerman et al., 1997) among others.

There is no single model or method in terms of complexity, cost, precision, time, and robustness, which can be regarded as best method to be used in network faults modeling. Recent approaches lean towards a combination of two or more methods together to solve complex problems (like cellular networks systems) (Walrand et al., 2000), (Heckerman et al., 1997), (Frey et al., 1997). The choice of a method would depend on a specific problem case. However, the following factors may be considered: (1) Implementation complexity, (2) Facility for construction of a theoretical model of the *object network*, (3) Performance, (4) Facility to adapt to change in the object network, and (5) Precision. Network faults prediction must take into account the object network characteristics, which can be looked at from technology viewpoint i.e., CDMA, FDMA, GSM, UMTS, WCDMA, CDMA2000 (Frey at al., 1997), GPRS, EDGE, DECT, etc.

The nature of application area will also dictate the choice of a method to adopt. The high cost of implementation and adaptation to changes in the object network make it difficult to apply rule-based approaches in large cellular networks and hence are better used in network elements whose configuration is rarely altered. The other approaches that are less sensitive to changes in the object network i.e., case-based approaches still lack a theoretical basis, which would allow their utilization in large, size commercial cellular service network systems.

The complexity of the problem to be solved sometimes brings the *exceptions*, which can be effectively treated by mechanisms that are used for the implementation of *non-monotonic* reasoning (Williamson et al., 2005). This means each identified exception requires the reformulation of the already established rules or the creation of new ones at the development phase. The complexity of the solution increases, which reduces the performance and robustness. Rule-based systems provide additional structuring which facilitates the development of applications but because of the tendency to reduce performance, they are not attractive for the implementation of more complex cellular network systems.

The complexity of cellular network faults prediction problem makes it extremely difficult to obtain exact solutions (Donini et al., 1990). This brings uncertainty as a factor in fault prediction process that needs to be considered. Fuzzy logic, Bayesian networks (cf. Section 2.6), case-based reasoning and artificial neural networks are some of the approaches that can deal with uncertainty. Each of these alternatives have got advantages and disadvantages, for example, defenders of fuzzy logic based approaches argue that they simplify applications development and result in working products with excellent performance; on the other hand it is tuning (like membership functions) is very hard and it lacks a solid mathematical support which hinders its adoption in a larger number of applications.

Bayesian network based methods, which were first utilized in 1921 in the analysis of harvesting results (Katzela et al., 1996) count on a solid mathematical support. They win more acceptances in the community of computing scientists as a suitable option to the solution of problems involving uncertainty (Katzela et al., 1996). These factors contributed to the adoption of Bayesian networks in this work.

2.6 Bayesian networks

Bayesian networks model is named after Thomas Bayes, who proved a special case of what is called Bayes' theorem. The term Bayesian, however, came into use only around 1950 (Heckerman, 1997). It provides an approach to the treatment of uncertainty with incomplete and inaccurate available data to produce inferences.

A Bayesian network is a Directed Acyclic Graph (DAG) in which each node represents a random variable to which conditional probabilities are associated, given all the possible combinations of values of the variables represented by the directly preceding nodes; an edge in this graph represents conditional probabilities between the variables corresponding to the interconnected nodes (Meira, 1997; Heckerman, 1997).

The terms subjective probability, personal probability, epistemic probability and logical probability describe some of the schools of thought, which are customarily called "Bayesian". These terms overlap but there are differences of emphasis. A subjective probability expresses the degree of belief of an expert related to the occurrence of a given event, based on the information this person has available up to the moment. The use of subjective probabilities is very often the only resource in situations where analytical or experimental data is very hard or even impossible to obtain.

It is possible sometimes to evaluate conditional probabilities from empirical data obtained from the past behaviour of the network service provider under study. Given a Bayesian network and a set of evidences it is possible to evaluate the network, that is, to calculate the conditional probability associated with each node, given the evidences observed up to the moment. Generally speaking, this is a NP-hard problem (Chickering et al., 2004) but with the use of appropriate heuristics and depending on the problem dealt with; networks containing thousands of nodes may be evaluated in an acceptable time. An example of a Bayesian network of faults prediction is shown in Fig. 2.

Why Bayesian Networks?

The main reasons why Bayesian networks was chosen for cellular network faults prediction include (Hood et al., 1997a), (Hood et al., 1997b), (Hood et al., 1998):

- *Mathematical support*: the Bayesian networks count on a solid mathematical support, which allows the analysis of the model in view of the knowledge of its performance and precision before an implementation is carried out.
- *Robustness*: approximate answers can be obtained, even when the existing information are incomplete or imprecise whenever new information become available, the Bayesian networks allow a corresponding improvement in the precision of the correlation results.
- The facilities are readily available for the construction of the Bayesian network.

- Bayesian networks have the capacity to identify, in polynomial time, all the conditional independence relationships that are extracted from the information gained by the Bayesian network structure.
- The capacity for non-monotonic reasoning, through which previously obtained conclusions may be withdrawn as a consequence of the knowledge of new information.
- The capacity to carry out inferences on the present state of telecommunication networks from the combination of: a) statistical data empirically surveyed during the network functioning, b) subjective probabilities supplied by specialists, and c) information (that is, "evidences" or "alarms") received from the telecommunications network, in real time.
- It is simple and effective.

Concerns about Bayesian Networks

Although the use of Bayesian network for knowledge generation from data produces good results on some benchmark data sets, there are still some concerns. These include (Heckerman, 1996, 1999):

- *Node ordering requirement* - many Bayesian network learning algorithms require additional information, which is mostly an ordering of the nodes to reduce the search space. Unfortunately, this information is not always available.
- *Computational Complexity* - practically all Bayesian network learners are slow, both in theory and in practice. For example, most dependency-analysis based algorithms require an exponential numbers of "conditional independence" tests.
- *Lack of publicly available learning tools* - although there are many algorithms for this learning task, very few systems for learning Bayesian networks systems are publicly available. Even fewer systems can be applied to real-world data-mining applications where the data sets often have hundreds of variables and millions of records.

2.7 Cellular network faults modelling process

The cellular network fault modeling process consists of five independent processes. These processes are indefinitely repeated and take into account the existence of the assumptions outlined in Section 2.8. These include:

- Network fault alarms acquisition by the network management system
- Classification of the cellular network fault alarms received, according to time windows and originating network element.
- Correlation at network element level, from network fault alarms originally generated by the network elements or obtained through other correlation processes, depending on the correlation topology adopted. The existence of this process is not mandatory at a first moment, as it can be gradually implemented in each network element, according to necessity and taking into consideration peculiarities of each one of them.
- Random variables corresponding to each network element are updated, according to the state of these network elements, which is given by the network fault alarms received and by the result of the correlations carried out on these fault alarms.
- Network fault alarm correlation at the cellular network level, through the evaluation of the new probabilities associated to the fault states defined for each network element, in view of the evidences available at each moment.

2.8 Assumptions

For the models to perform some of the functions there were assumptions made as a condition for the model application in this work. These include:

- It was assumed that it is possible to define a discrete random variable values representative of the state for each network element corresponding to a node in the graph of the cellular network model.
- There is an integral network management system that collects information, from which values are attributed, in real time, to the variables mentioned at element level.
- The managed cellular network service provider is modeled in conformity with the ITU-T rules.
- The prior probabilities related to the variables of each root node, and the local probabilities related to the variables of the other nodes, may as well be alternatively attributed to:
  - through the *relative frequencies* of the corresponding events which are calculated from the data collected by the network management system
  - by using *relative likeliness*. This consists of estimating the probabilities from the subjective judgment of an expert. This method will be useful whenever there is not enough data to permit the estimation of the relative frequencies, which may occur due to the low frequency of the phenomena observed, or even due to the nonexistence of sufficient network management resources.

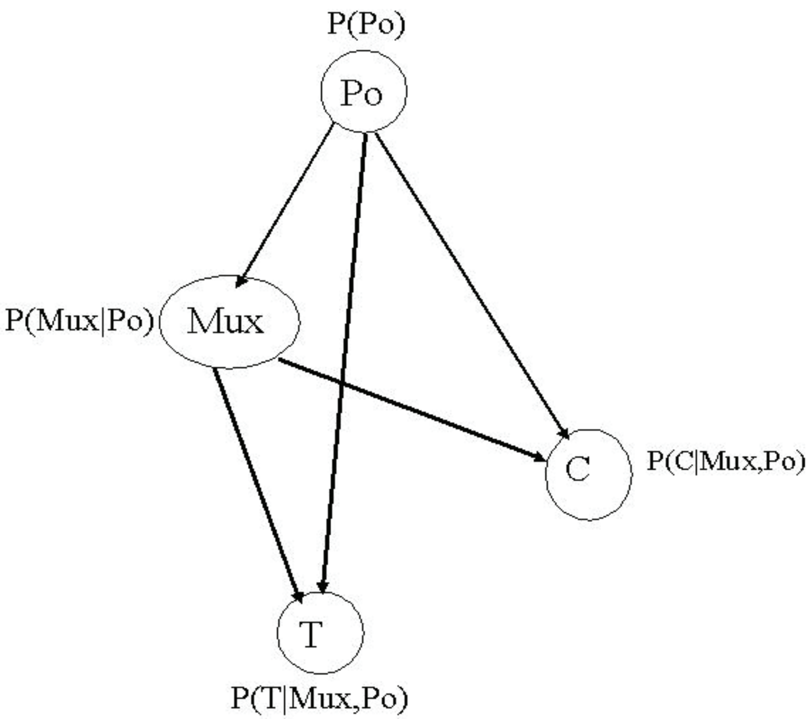


Fig. 2. A Bayesian Network of faults prediction

3. Mobile intelligent agents

3.1 Definition and related work

An Intelligent Agent (IA) is defined as "software that assists people and acts on their behalf. Intelligent agents work by allowing people to delegate work that they could have done, to the software agent. Agents can, just as assistants can, automate repetitive tasks, remember things you forgot, intelligently summarize complex data, learn from you, and even make recommendations to you" (Gilbert, 1997). In this work, we define an IA as an autonomous program with the capability of controlling its own actions and decision-making based on prior knowledge, past experience and on its perception of its environment in pursuit of predefined goals. However, these kinds of agents cannot migrate from one host to another making them undesirable for this work. Mobile Intelligent Agent (MIA), on the other hand, have ignited interest among researchers in Information Communication Technology (ICT) with applications as diverse as e-commerce, computer games, interface design, personalised information management and management of complex cellular networks.

Proactive anomaly detection using distributed intelligent agents was proposed in (Thottan & Ji, 1998) and faults prediction at the network layer using intelligent agents was proposed in (Thottan & Ji, 1999). While these works explored ways of applying intelligent agents and Bayesian Belief Network (BNN) in cellular network faults prediction, they did not address the relationship between cellular network faults and service provision, and only explored the network layer. Pissinou et al (Pissinou et al., 2000) apply mobile agents to automate the fault management in wireless and mobile networks. While this work delves into detection of faults and automated recovery from faults, it does not deal with reporting such faults before they occur. Andrzej Bieszczad et al (Bieszczad et al., 1998) discussed the potential uses of mobile agents in network management and in Elerftheriou and Galis (Eleftheriou & Galis, 2000) MIAs are explored for network management systems. While both papers discuss the application of mobile agents in fault analysis, they fall short of predicting such faults and relating them to cellular network services. Thottan and Ji (Thottan & Ji, 1998, 1999) presents a proactive anomaly detection using distributed intelligent agents and faults prediction at the network layer using intelligent agents. They used intelligent agents, throbbing technique and Bayesian Belief Network (BNN) in network faults detection and prediction. The deployed agents obtain relevant MIA data, provide temporally and spatially correlated predictive alarms, and time correlated abnormal changes in the individual MIA variables. Their testing showed successful prediction rate of seven faults out of nine faults with a prediction time in the order of minutes. However, the authors failed to relate faults to network services. In the present work we address this relationship.

We define Mobile Intelligent Agent (MIA) as a software program that acts on behalf of a user or another program and is able to migrate from one network node to another in a network system under its own control. The MIA chooses when and where it will migrate and may interrupt its own execution and continue elsewhere on the network. The agent returns results and messages in an asynchronous fashion. We exploit the intelligence and ability to cooperate features of MIA in this work.

3.2 Why Mobile Intelligent Agents?

Almost every task that can be performed by MIA can be done by stationary intelligent agents. However, the use of MIA brings certain benefits over other technologies such as stationary intelligent agents, remote objects, etc, including (Chess et al., 1998; Bieszczad et al., 1998; Eleftheriou & Galis, 2000):

- Efficiency savings – CPU consumption is limited, because a mobile agent executes only on one node at a time. Other nodes do not run an agent until needed.
- Space savings – Resource consumption is limited, because a mobile agent resides only on one node at a time. In contrast, static multiple servers require duplication of functionality at every location. Mobile agents carry the functionality with them, so it does not have to be duplicated.
- Reduction in network traffic – Code is very often smaller than data that it processes, so the transfer of mobile agents to the sources of data creates less traffic than transferring the data. Remote objects can help in some cases, but they also involve marshalling of parameters, which may be large.
- Asynchronous autonomous interaction – Mobile agents can be delegated to perform certain tasks even if the delegating entity does not remain active.
- Interaction with real-time systems – Installing a mobile agent close to a real-time system may prevent delays caused by network congestion.
- Robustness and fault tolerance – If a distributed system starts to malfunction, then mobile agents can be used to increase availability of certain services in the concerned areas. For example, the density of fault detecting or repairing agents can be increased. Some kind of meta-level management of agents is required to ensure that the agent-based system fulfils its purpose.
- Supports for heterogeneous environments – Mobile agents are separated from the hosts by the mobility framework. If the framework is in place, agents can target any system. The costs of running a Java Virtual Machine (JVM) on a device are affordable.
- On-line extensibility of services – Mobile agents can be used to extend capabilities of applications, for example, providing services. This allows for building systems that are extremely flexible.
- Convenient development paradigm – Creating distributed systems based on mobile agents is relatively easy.
- Easy software upgrades – A mobile agent can be exchanged virtually at will.
- Dynamic adaptation – mobile agents can sense their environment and react autonomously to changes and can distribute themselves among hosts in the network to maintain optimal configuration. In the case of a mobile agent moving across a number of host nodes, it can adapt its future behaviour according to information that it has already collected and stored in its state.

However, MIA still suffer from certain cost constraints, which include (Chess et al., 1998; Eleftheriou & Galis, 2000): migration and machine load overhead; high costs in speed for Remote Method Invocation (RMI); resource management; standardization and interoperability; and lastly the directory service used by the agents seemed to slow down communication when the number of agents increases.

4. The mobile intelligent agent model

4.1 Proposed mobile architecture

The proposed MORSBOSS mobile intelligent model is designed into three tiers conducive to real-time processing. The first tier is specific network elements, which are within the cellular network environment. The mobile agent monitors the network elements/nodes in this environment. The next tier where MORSBOSS agent operates is the mobile agency. It mediates between the managed network elements and the data tier. The MORSBOSS agent analyses and then logs the fault alarms into the database. Data is the last tier where generated alarms (evidence of fault occurrence) are stored. The alarms variables are aggregated and computed based on BNM and then these alarms are combined using a priori information about the relationships between the variables to produce network element alarms, which are the indicators of the network health as shown in Fig. 3.

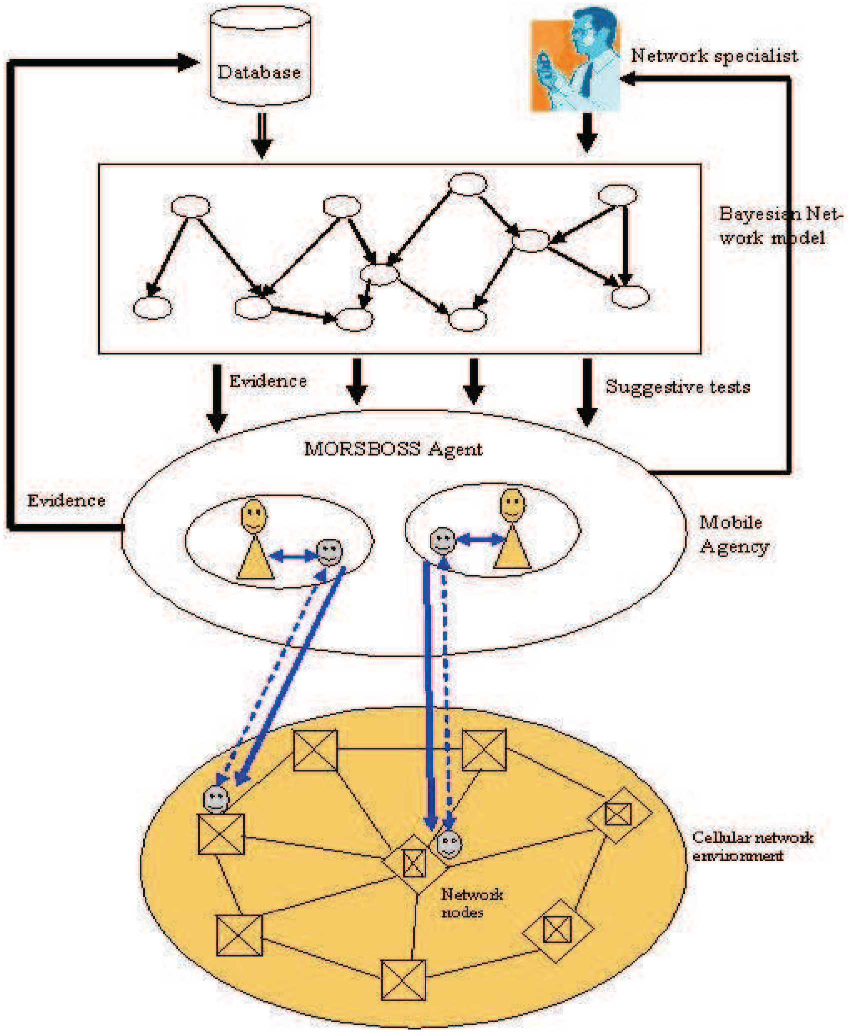


Fig. 3. Proposed MORSBOSS Architecture

4.2 Why cellular network faults prediction?

Faults prediction brings a number of benefits to the cellular network service providers. Some of these include (Hood & Ji, 1997a, 1997b, 1998; Danyluk & Provost, 2002):

1. Accurate fault prediction models support project planning and steering.
2. Faults prediction helps network managers in re-routing of network traffic. Network managers can take corrective action before the faults occur, thereby ensuring services reliability and availability over the network.
3. Decision making, i.e., whether to buy from a particular vendor or not, whether to buy a particular hardware or not, etc. It may also be decided that for elements with a high-predicted fault-proneness, say, above 25%, the element design shall undergo quality assurance (QA) activities such as inspections, extensive unit testing, etc. QA tests help to improve the quality of system with each fault that is discovered and corrected.
4. Operations cost will be minimized if network faults are found as soon as they occur. Faults which are discovered early are cheaper to repair and hence such a scenario leads to offering of cheap and reliable services.
5. The fault prediction models provide a mapping from a hard to interpret design measurement data to easily interpreted external quality data.
6. Fault prediction models provide a sound method to combine multiple factors into one cohesive model i.e., take into account the various factors that make certain network elements fault-prone.
7. Highly accurate fault prediction models can be beneficial when highlighting trouble areas in cellular network system.

Uncertainty Causing Factors

Cellular networks being dynamic in nature and as has been demonstrated in this chapter, uncertainty is inherently associated with the cellular network faults prediction process. This uncertainty is largely driven by the possibility of including erroneous data. There are basically four main sources of data errors. These are:

- The influence of factors not captured by the managed system model
- The imprecision in the calculation of the probabilities distribution values
- The imprecision in the capture and transference of the alarms.
- Imprecision in the information obtained from other correlating processes.

4.3 Cellular network faults prediction models

The cellular network under study has power (Po), cell (C), transmission (T), and multiplexer (Mux) faults as the network variables to be estimated as shown in Fig. 4. Each variable has observations, which are stored in the database. The arrows indicate cause and effect of the cellular fault. The structure was chosen from the natural grouping of the network dynamics into four variables mentioned above. The data from different database variables were combined through the probabilistic framework defined by the Bayesian belief network, where the probabilities were estimated from the network variables as in (Kogeda & Agbinya, 2006; Kogeda et al., 2006). These include the conditional probability, for example, assuming multiplexer (Mux) is ok then the only reason it may not perform its functions is when power (Po) fails. This can be calculated by equation (1) and the joint probability distribution using equation (2).

$$p(Mux \mid Po) = \frac{p(Mux) \times p(Po \mid Mux)}{p(Po)} \tag{1}$$

$$\mathrm{p(Po,Mux,C,T) = p(Po) \times p(Mux) \times p(C \mid Po,Mux) \times p(T \mid Mux)} \tag{2}$$

The prediction factor, 'belief' that a variable $X_k \notin \{X_m,\ldots,X_p\}$ assumes the value x_k is computed using equation (3) when one knows a set of evidences $e = \{X_m = x_m,\ldots,X_p = x_p\}$, constituted by all the known values of the random variables of Bayesian network, where $\{X_m,\ldots,X_p\} \subset X = \{X_1, X_2,\ldots,X_n\}$:

$$p(X_k = x_k \mid e) = \frac{p(X_k = x_k) \times p(e \mid X_k = x_k)}{p(e)} \tag{3}$$

The above equations (1, 2 & 3) form part of the MIA engine, which is able to foresee and move to a node likely to be faulty. The proposed model in this paper utilizes different artificial intelligent techniques. The development of the proposed application (software) requires knowledge extraction from the alarms stored in the database (MORSBOSS database), which must be updated with the new knowledge that might have been discovered from the alarm data and the belief of foreseen fault occurrence.

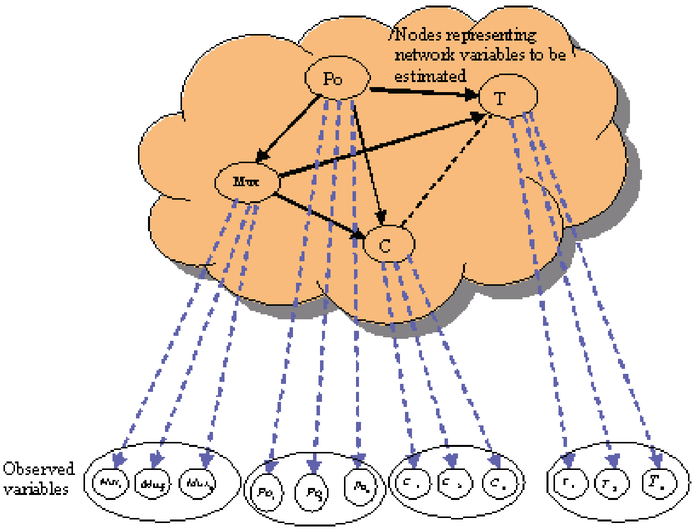


Fig. 4. A Bayesian network for fault prediction

The MORSBOSS database developed in MySQL stores the variable alarms, which are grouped into four distinct fault variables using probabilistic framework defined by Bayesian belief network. Computation of conditional and joint probabilities based on the information at hand is done within MORSBOSS agent using equations (1) and (2). It then interprets the computed figures, which may lead to evidence of an eminent fault and suggestive tests to be done in order to prevent the eminent failure. It is at this point that MORSBOSS agent sends alarm message to the engineering assistant agent about the health of the network. This message can be in the form of excellent, good, fair or critical. Whenever a critical alarm is sent to the engineering assistant agent who in turn informs selected customers (based on profile) about the foreseen fault, the network is at a high risk of failure. Depending on time to fault, the MORSBOSS agent can decide to inform the selected customers and engineering assistant agent at the same time.

Besides informing the engineering assistant agent, the MORSBOSS Agent moves with speed to the node, which is deemed to become faulty. Since it is a mobile agent, it clones itself in order to cooperate and do the task at hand. In this manner, the agent is able not only to predict but also to be ready to resolve the error; for example, the traffic can be re-routed to other routes avoiding the faulty node. Once a fault occurs then the agent communicates the message (i.e., start time, end time, node id, service id, etc) as evidence, which is sent to the database as shown in Fig. 3.

The interpretation of the computed values may show no change, positive change or negative change. The agent will determine this change within a given time window. We took average time to fault as our time window for repeating this computation in order to determine the node with high frequency failure.

MIA operates in an environment, which is cellular network node. The nodes often go into different states of operation. These states can be normal or abnormal. When a state of a node is normal, then such a node is assumed to be operating well without errors and abnormal state implies that the node is faulty. Let us assume that the node (N) may be in finite state set E of discrete states: $N = \{e, e', ...\}$ Where e is normal state and e' is abnormal state. The MIA is assumed to have an array of possible actions available to them, which they can perform to transform the state of the node. The state of the node also dictates which action the MIA will perform. The action of MIA is also dependent on the history of the node. State transformer function (JADE-LEAP website) is used to represent the effect that MIA's action have on the node:

$$\tau : R^a \rightarrow p(N) \tag{4}$$

Where R is a run of MIA in an environment; a actions.

We compute the likelihood value of a fault occurring using Bayesian network model. Using MIA, M and our modeling techniques, we define the likelihood of an environment being in abnormal state by:

$$M = \begin{cases} p(e_0, e_1, e_2, ...e_n) & \\ 0 & otherwise \end{cases} \tag{5}$$

Where $e_0 \in N$ is the initial state of the node.

The Utility (U) function of the MIA can be measured on some particular runs (r) using:

$$U(r) \cong \frac{fr}{fd} \tag{6}$$

Where fr is number of faults fixed in r; fd is number of faults that appeared in r.

Let us write $p(r \mid M, E)$ to denote the probability that run r occurs when MIA m is placed in environment E, clearly as:

$$\sum_{r \in R(M,E)} p(r \mid M, E) = 1. \tag{7}$$

Then the optimal MIA $Mopt$ in an environment E is defined as the one that maximizes expected utility:

$$M_{opt} = \underset{M \in m}{\arg\max} \sum_{r \in R(M,E)} U(r)p(r \mid M,E) \tag{8}$$

5. Implementation architecture

This work has been implemented using Java Agent Development Framework and Lightweight Extensible Agent Platform (JADE-LEAP) (Moreno et al., 2002; JADE-LEAP website, 2007), a mobile agent platform distributed by Telecom Italia (Moreno et al., 2002; JADE website, 2007).

5.1 Why JADE architecture?

JADE (JADE website) was chosen for a combination of benefits offered including (Altmann et al., 2001):

- Simplicity in usage and agent programming;
- good online community support and documentation;
- support for the Foundation for Intelligent Physical Agents (FIPA) (FIPA Specification, 2007) standards;
- efficient and tolerant of faulty programming and
- it is open source (free), among others.

However, it is worth noting that JADE does not provide support for migration between different execution environments as a drawback.

JADE has already been integrated into different major architectures, such as J2EE and .NET allowing JADE to execute multi-platform proactive applications. JADE can take J2EE for servers, J2SE (Fagin et al., 1995) for PCs and desktop computers, Personal Java (Pjava) for wireless mobile devices supporting Pjava (i.e., Personal Digital Assistant (PDA)) and J2ME with Connected Limited Device Configuration (CLDC) and Mobile Information Device Profile (MIDP) for mobile devices supporting MIDP (i.e., cell phones) as shown in Fig. 5.

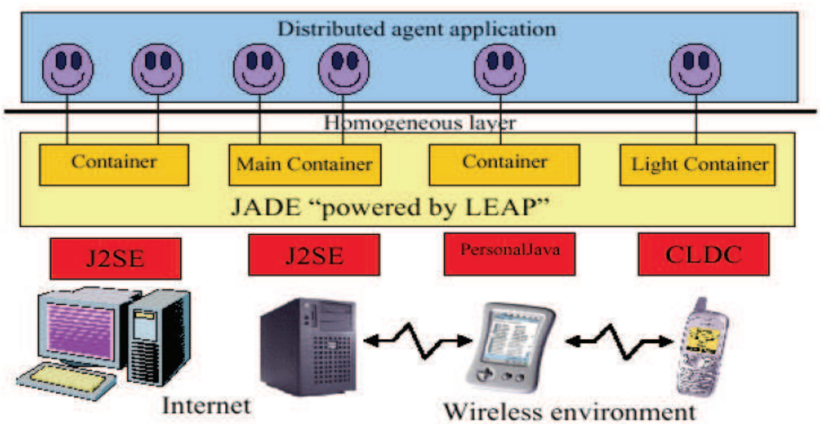


Fig. 5. JADE-LEAP agents' architecture, from JADE-LEAP user's guide

A MIA agent is a JADE-LEAP agent, which is a mobile agent with autonomy and high degree of collaboration. The system automatically creates a new Client Agent to act on behalf of the user once a new user logs onto the system. The agent name is unique, and based on the user ID. The Client Agent is able to obtain and maintain the user's personal information and preferences, and to react to various incoming and outgoing messages and requests intended for the user. The Client Agent is automatically removed from the system once the user log outs.

6. Experimental results

We set up a simple wireless network of eight devices. 2 iPAQs, Router, 2 desktop PCs, 2 laptops and an access point are the devices that act as nodes in our simple wireless network. The implemented MIA is then executed from the desktop-1, which acts like host to the MIA. The PCs, laptops and iPAQs are connected to the network wirelessly via the wireless access point as shown in Fig.6. We decided to send simple messages, which arrived in time as any normal healthy cellular network service provider.

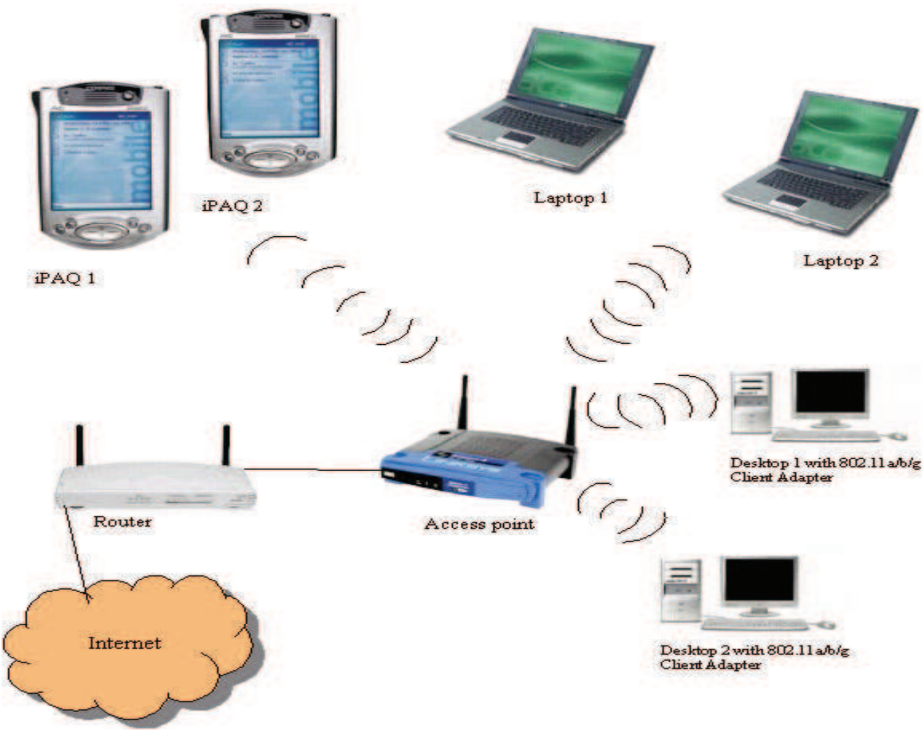


Fig. 6. Wireless network for Experiment

In our testing of the models, we injected faults to various nodes after a random number of minutes. The injection was done with a view to see if the MIA could respond to not only detecting the faults but also in predicting the faults before they occurred. We made 400 runs over a period of three months. A run took an average of 30 minutes. During each run the MIA would update the 'belief' after a time window of 5 minutes, by logging in the faults and computing the new 'belief' of each and every variable. The node status is stored in the database and bears a typical format (for example, faultID: 1 serviceID: 1 faultName: Cell prob: 0.0898438 faultState: VERY UNLIKELY).

The customers consume services, whose performances are affected by the fault. In case of a fault, the MIA may inform the customer of a fault as well as network engineer. The typical message to the customer who consumes the service affected is (Dear Customer, Fault ID: 1 has been created for service ID: 1 MORSBOSS will endeavour to resolve the problem ASAP. Please do not reply this SMS).

We injected faults to the network during each run of the MIA. Most of the network faults were injected automatically. Ms Visual Basic for application program was written to automatically put off the devices at random time intervals, VBScript for injecting transmission fault, and manually moving portable devices away from the wireless access point thereby leading to loss of signal fault, among others. In our experiments of 400 runs, our model could report 316 faults before they occurred, 38 faults were reported after they had occurred, 26 faults could not be detected, 20 faults alarms were false, giving us a success rate of about 79% prediction rate, 9.5% delayed results, 6.5% no results, and 5% false alarms. The probability of the network being faulty in each of the runs we made is shown in Fig. 7. The utility of the MIAs is shown in Fig. 8.

7. Conclusion

The model utilizing different artificial intelligent techniques was proposed in this Chapter. Cellular network faults prediction using mobile intelligent agent technology and Bayesian belief networks was presented. The experimental results show a prediction rate of 79% with 5% false alarms. The research will strive to improve on the success rate and incorporate swarm intelligence in furthering the cellular network fault prediction.

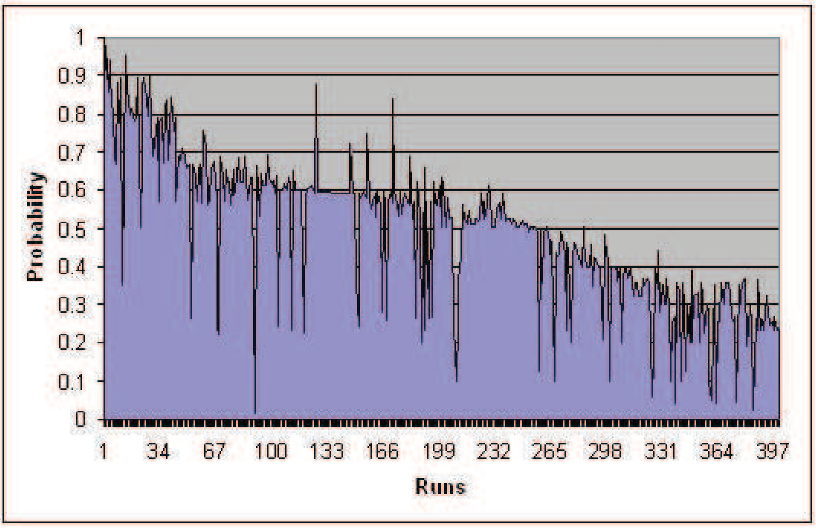


Fig. 7. Probability of faults in each run

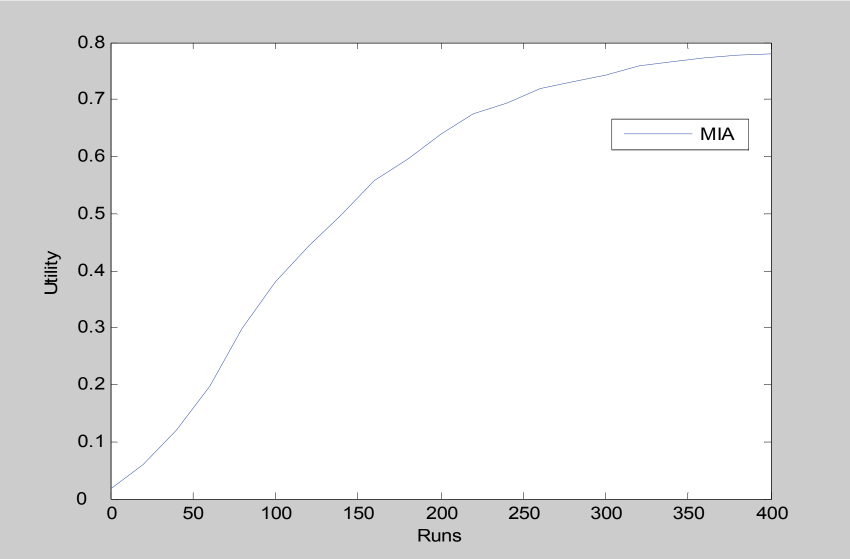


Fig. 8. MIAs Utility

8. Acknowledgement

The authors would like to thank the National Research Foundation, South Africa for the financial support that has made this work possible

9. References

Agnar, A. & Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *Artificial Intelligence Communications* 7, no. 1, pp. 39-52, 1994.

Altmann, J.; Gruber, F.; Klug, L; Stockner, W. & Weippl, E. (2001). Using Mobile Agents in Real World: A Survey and Evaluation of Agent Platforms, *Workshop on Infrastructure for Agents, MAS and Scalable MAS, at the 5th International Conference on Autonomous Agents*. ACM Press, Montreal, Canada, pp. 33-39, June 2001.

Bieszczad, A.; Pagurek, N. & White, T. (1998). Mobile Agents for Network Management, *IEEE Communications Surveys and Tutorials*, Vol. 1(1), 1998.

Chess, D. M.; Harrison, C. G. & Kershenbaum, A. (1998). Mobile Agents: Are They a Good Idea?, IBM Research Report, April 27-30, 1998.

Chickering, D.; Heckerman, D. & Meek, C. (2004). Large-Sample Learning of Bayesian Networks is NP-Hard, *Journal of Machine Learning Research.* V5:pp.1287-1330, 2004.

Corkill, D. D. (1991). Blackboard Systems, *AI Expert* 6(9): pp.40 – 47, Sept.1991.

Danyluk, A. & Provost, F. (2002). Telecommunications Network Diagnosis, *In W. Kloesgen and J. Zytkow* (eds.), *Handbook of Knowledge Discovery and Data Mining,* Oxford University Press, 2002.

Donini, F. M.; Lenzerini, M.; Nardi, D.; Pirri, F. & Schaerf, M. (1990). Non-monotonic reasoning, *Artificial Intelligence Review*, Vol. 4(3), pp.163-210, 1990.

Duda, R. O.; Hart, P. E. & Stork, D. G. (2001). Pattern classification (2nd edition), Wiley, 2001, ISBN 0471056693.

Eleftheriou, G. & Galis, A.(2000). Mobile Intelligent Agents for Network Management Systems, *Proceedings London Communication Symposium*, 2000.

ETSI Guide (2001), Final drafts ETSI EG 202 009-3 V1.1.0 (2001-12) User Group; Quality of telecom services; Part 3: Template for Service Level Agreements (SLA). 2001.

Fagin, R.; Halpern, J. Y.; Moses, Y. & Vardi, M. Y. (1995). Reasoning about Knowledge, MIT Press, Cambridge, MA, 1995.

FIPA Specifications., Available from: http://www.fipa.org/

Frey, J. & Lewis, L. (1997). Multi-level reasoning for managing distributed enterprises and their networks, *In Integrated Network Management V*, pp.5-16, 1997.

Gilbert, D. (1997). Intelligent Agents: The Right Information at the Right Time, White Paper, *IBM Corporation*, 1997.

Hajji, B. & Far, B. H. (2001b). Continous Network Monitoring for Fast Detection of Performance Problems, *Proceedings of 2001 International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2001.

Hajji, B.; Far, B. H. & Cheng, J. (2001a). Detection of Network Faults and Performance Problems, *Proceedings of the Internet Conference*, Osaka, Japan, Nov. 2001.

Heckerman, D. (1996). Bayesian Networks for Knowledge Discovery, *Advances in Knowledge Discovery and Data Mining*, pp.273-305, 1996.

Heckerman, D. (1999). A Tutorial on Learning with Bayesian Networks, *In Learning in Graphical Models*, pp. 301-354, MIT Press, 1999.

Heckerman, D.; Meek, C. & Cooper, G. (1997). A Bayesian Approach to Causal Discovery, *Technical Report*, MSR-TR-97-05, Microsoft Research, Feb. 1997.

Holst, A. (1997). The Use of a Bayesian Neural Network Model for Classification Tasks, *Thesis, Studies of Artificial Neural Systems*, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, S-100 44 Stockholm, Sweden, September 1997.

Hood, C. S. & Ji, C. (1997a). Proactive Network Fault Detection, *Proceedings of the IEEE INFOCOM*, pp. 1139-1146, Kobe, Japan, April 1997.

Hood, C. S. & Ji, C. (1997b). Automated proactive anomaly detection, *In Integrated Network Management V*, California, USA, Vol.86, pp.688-699, 1997.

Hood, C. S. & Ji, C. (1998). Intelligent Agents for Proactive Fault Detection, *IEEE Internet Computing*, Vol. 2(2), pp. 65-72, March/April 1998.

JADE website., Available from: http://jade.tilab.com/

JADE-LEAP website., Available from: http://leap.crm-paris.com/

Katzela, I.; Bouloutas, A. T. & Calo, S. (1996). Comparison of distributed fault identification schemes in communication networks, *Technical report*, *IBM Corp.*, T.J. Watson Research Center, Yorktown Heights, NY, USA, January 1996.

Khardon, R. & Roth, D. (1997). Defaults and Relevance in Model Based Reasoning, *Artificial Intelligence*, Vol. 97, Number 1-2, pp. 169-193, 1997.

Khardon, R.; Mannila, H. & Roth, D. (1999). Reasoning with Examples: Propositional Formulae and Database Dependencies, *Acta Inf*, 36(4), pp.267–286, 1999.

Klaus-Dieter, A.; Bergmann, R. & Branting, L. K. (1999). Case-Based Reasoning Research and Development, *Proceedings of the Third International Conference on Case-Based Reasoning*, Berlin, Germany, July 27-30,1999.

Kogeda, O. P. & Agbinya, J. I. (2006a). Prediction of Faults in Cellular Networks Using Bayesian Network Model, *Proceedings of 1st IEEE International Conference on Wireless Broadband and Ultra Wideband Communications (AUSWIRELESS 2006),* pp. ISBN: 0-9775200-0-5, Sydney Australia, March 13-16, 2006.

Kogeda, O. P.; Agbinya, J. I. & Omlin, C. W. (2006b). A probabilistic Approach to Faults Prediction in Cellular Networks, *Proceedings of the 5th International Conference on Networking (ICN2006),* ISBN: 0-7695-2570-9, Mauritius, April 23-28, 2006.

Langley, P. & Simon, H. A. (1995). Applications of Machine Learning and Rule Induction, *Communication ACM* Vol. 38(11): pp. 54 – 64, November 1995.

Lazar, A.; Wang, W. & Deng, R. (1992). Models and algorithms for network fault detection and identification: A review, *Proceedings of IEEE ICC*, Singapore, pp.999-1003, November 1992.

Lin, Y. & Druzdzel, M. J. (1997). Computational Advantages of Relevance Reasoning in Bayesian Belief Networks, *Proceedings of the Thirteenth Annual Conference in Uncertainty in Artificial Intelligence (UAI-97)*, pp. 342-350, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1997.

Meira, D. M. (1997). A Model for Alarm Correlation in Telecommunications Networks, *PhD Thesis*, Federal University of Minas Gerais, Belo Horizonte, Brazil, Nov. 1997.

Mitchell, T. (1997). Decision Tree Learning, in T. Mitchell, *Machine Learning*, The McGraw-Hill Companies, Inc., pp. 52-78, 1997.

Moreno, A.; Valls, A. & Viejo, A. (2002). Using JADE-LEAP to implement agents in mobile devices, 2002.

Pissinou, N.; Bhagyavati & Makki, K. (2000). Mobile Agents to Automate Fault Management in Wirelss and Mobile Networks, *IPDPS Workshops*, Lecture Notes in Computer Science, Vol. 1800, 2000, pp. 1296-1300, Springer Verlag.

Thottan, M. & Ji, C. (1998). Proactive Anomaly Detection Using Distributed Intelligent Agents, *IEEE Network*, Sept./Oct. 1998.

Thottan, M. & Ji, C. (1999). Fault Prediction at the Network Layer Using Intelligent Agents, *Proceedings of IEEE/IFIP Integrated Network Management*, May 1999, Boston, MA.

Walrand, J. & Varaiya, P. (2000). High-performance Communication Networks, *Second edition*, Morgan Kaufmann, 2000.

Watson, I. (1997). Applying Case-Based Reasoning: Techniques for Enterprise Systems. *Morgan Kaufmann*, July 1997, ISBN: 9781558604629.

Widrow, B. (1989). DARPA Neural Network Study, *AFCEA International Press*, 1989.

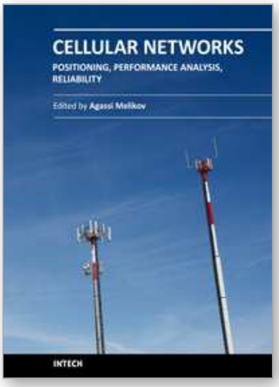
Williamson, C.; Halepovic, E.; Sun, H. & Wu, Y. (2005). Characterization of CDMA2000 Cellular Data Network Traffic, *LCN*, pp.712–719, Nov. 2005.

Winston, P. (1992). Learning by Building Identification Trees, in P. Winston, *Artificial Intelligence*, Addison-Wesley Publishing Company, pp. 423-442, 1992.

Wooldridge, M. J. & Jennings, N. R. (1995). Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review,* Vol. 2, 1995, pp. 115-152, ISSN

Zadeh, L. A. (1962). In the engineering journal, *Proceedings of the IRE*, 1962.

Zadeh, L. A. (1965). Fuzzy sets, *Information and Control*, pp. 8:338-353, 1965.



Cellular Networks - Positioning, Performance Analysis, Reliability
Edited by Dr. Agassi Melikov

ISBN 978-953-307-246-3
Hard cover, 404 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Wireless cellular networks are an integral part of modern telecommunication systems. Today it is hard to imagine our life without the use of such networks. Nevertheless, the development, implementation and operation of these networks require engineers and scientists to address a number of interrelated problems. Among them are the problem of choosing the proper geometric shape and dimensions of cells based on geographical location, finding the optimal location of cell base station, selection the scheme dividing the total net bandwidth between its cells, organization of the handover of a call between cells, information security and network reliability, and many others. The book focuses on three types of problems from the above list - Positioning, Performance Analysis and Reliability. It contains three sections. The Section 1 is devoted to problems of Positioning and contains five chapters. The Section 2 contains eight Chapters which are devoted to quality of service (QoS) metrics analysis of wireless cellular networks. The Section 3 contains two Chapters and deal with reliability issues of wireless cellular networks. The book will be useful to researches in academia and industry and also to post-gradute students in telecommunication specialitiies.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Okuthe P. Kogeda and Johnson I. Agbinya (2011). Automation of Cellular Network Faults, Cellular Networks - Positioning, Performance Analysis, Reliability, Dr. Agassi Melikov (Ed.), ISBN: 978-953-307-246-3, InTech, Available from: http://www.intechopen.com/books/cellular-networks-positioning-performance-analysis-reliability/automation-of-cellular-network-faults

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.