

Mobile Agents to Automate Fault Management in Wireless and Mobile Networks¹

Niki Pissinou, Bhagyavati, Kia Makki

The University of Louisiana at Lafayette
{pissinou/bxb8329/makki@cacs.usl.edu}

Abstract. This paper studies the automation of fault management of wireless and mobile networks. A network management protocol similar to SNMP with an integrated architecture using mobile agents will prove effective in fault management of such networks. In view of the above, we design a wireless network management protocol to support fault management. In particular, we use mobile agents to detect, diagnose and recover from faults in wireless and mobile networks.

1. Introduction

The main drawback of existing protocols for network management is the impracticality of having a human administrator at all times, working at the same pace, and for every node in the network. This shortcoming is especially arduous in the case of a large network with many users with different needs and expectations. Therefore, there is an acute need for automating some or, ideally, all of the network manager's responsibilities. This paper looks at automating the area of fault management, the detection of and recovery from faults. An important task of network administrators is fault management: the ability to detect faults, diagnose the cause(s) for the fault, and provide ways for the network to recover from them. In a typical wireless and mobile network, the nodes are constantly moving and are called mobile units.

This work looks at networks that have wireless capabilities as well as mobility. Such networks are expected to provide the same type of access and quality of services to customers on the move that customers using wireline networks now possess. The main problem addressed in this paper is the extension of the architecture of a wireless and mobile network by adding mobile agents to automate recovery from faults. Since SNMP is considered the standard network management protocol for the wireline networks of today [8], extending it to wireless and mobile networks and integrating mobile agents into this new architecture is bound to work as well, if not better, than wireline networks. Since there is no currently existing network management protocol

¹ Partially funded by NSF grant number DUE-9751414, NSF EPSCOR grant, NASA grant number NAG5-7127, BoRSF RD-A-39, ENH-TR-95, and LEQSF Industrial ties grant.¹

for such networks, an SNMP-like protocol would best serve the needs of customers because of inherent simplicity and widespread usage.

2. The Fault-Tolerant Wireless Network Management Architecture

This section looks at the architecture of a wireless and mobile network extended by using mobile agents in an effort to automate fault management. Dependability is the primary weakness of these networks [1]. The main liability of existing protocols for network management is the impracticality of having a human manager at all times. Therefore, there is an enormous need for automating network management. As networks become more and more complex, a human administrator is in danger of getting overwhelmed and not being able to meet the service needs. In such a scenario, automation of fault management in networks is a pressing necessity.

A brief description of the components of the architecture is presented. In this paper, we consider systems where the base stations are connected through wires, and the channel between the mobile units and base stations is wireless. If a mobile unit wants to communicate with another unit, it can do so via the underlying base stations. A mobile unit can have three states [2]: active (on), sleep, disconnected (off). Even while the mobile unit is off, control signals are still communicated to and from the base station. If the mobile unit is in the off state, then there is a waiting period until it goes to the on state. If it is in sleep state, then the super agent decides that there is no fault, and tries again after some time. We consider in this paper a network wherein only the communication between mobile unit and base unit is wireless.

3. Overall Description of Methodology

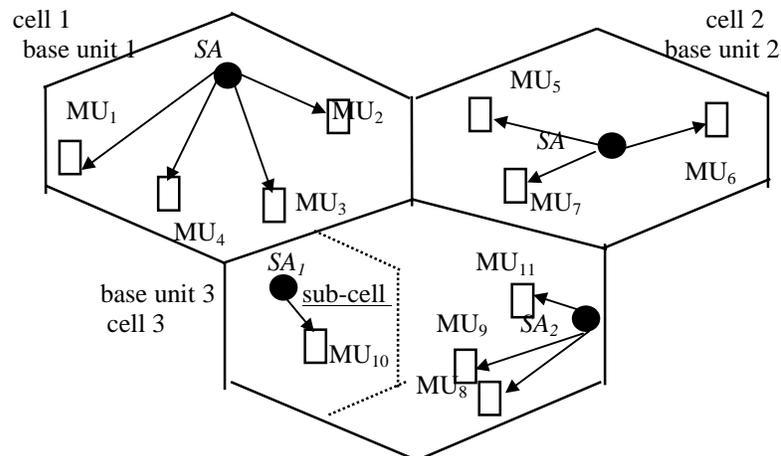


Figure 1: System Architecture

MU = Mobile Unit, SA = Super Agent, BU = Base Unit, MA = Mobile Agent

The super agent consists of three modules: history, diagnostic and recovery modules. The mobile agent reports to the super agent after copying the information from the mobile unit. If there was no fault found, then the super agent records that fact, along with the time, in the log, updates its VLR entry, and transmits the information to the home base unit, which updates its HLR entry. When a mobile unit, shown in the architecture, enters a cell, it notifies the base unit of its arrival. This is done by periodically transmitting short messages and judging the closest base unit by the fastest response time of its acknowledgment [2]. The base unit broadcasts this information to all super agents in the cell. The super agent of that sub-cell acknowledges and sends a mobile agent to the mobile unit. The mobile agent copies the parameter values of the mobile unit's profile and reports back to the super agent, which then performs a comparison with the history. If there is no mismatch or discrepancy, the super agent concludes that there is no fault and performs the necessary updates, including an update of this record in the history module.

If there is a mismatch, however, the possibility of a fault cannot be ruled out. The super agent records this fact in the history module and performs other updates (for example, in the VLR). The history module passes on the mismatched entry to the diagnostic module, and the super agent does a table lookup to find the type of fault that most likely caused the mismatch. Then the super agent passes these results to the recovery module as parameters, looks up the table and determines which procedures to initiate for recovery. The super agent then launches a repair agent, which migrates to the mobile unit and repairs the fault using the recovery procedure embedded in it.

4. Fault Management

A typical network management protocol provides information on the number of nodes in the network, which base stations are homes to which mobile units, and so on. This information aids in fault management. Typical procedures in fault management include the maintenance and periodic examination of error logs. The periodicity of the examination depends on the network traffic and the degree of robustness that the system is required to maintain. If the mobile agent is not "smart" enough to recover from or even detect the fault, then a human will be called in, who can recover from it. The human then updates the logs, which the super agent can use to learn from past behavior. The logs have information about the possible type of fault that led to the problem, and recovery from that fault, and how much time was taken for this procedure. Thus, after being operational for some time, the system can detect, diagnose and recover from any type of fault. The amount of time it takes a mobile agent to learn how to recover from the different types of commonly occurring faults depends on its speed, capacity and sophistication.

4.1 An Example of the Steps in Fault Correction and Recovery

In the fault recovery task, a mobile agent performs the necessary repair procedure. Hence it is called a repair agent. An example is considered next. Assume that the user informs the network manager of a problem in running an application.

The manager will retrieve the information about that node without even physically being present at the site. We can access diagnostic information such as utilization rate and system uptime at the network manager's workstation. For instance, the repair agent, disk and buffer usage can be checked to see if they are exceeded. Furthermore, this check admits of two kinds of failures: allocation failures and near-the-limits failures [7]. If this check unearths such a failure, then the repair agent, or disk or buffer should be erased, and thus recovery from this problem is possible.

If the repair agent, disk or buffer usage is not exceeded, a check is performed to see if there are too many users or too many processes. If either case were true, then clearly the CPU (Central Processing Unit) of that remote site/node is overloaded. Then recovery may be accomplished by killing some processes -- either the longest running or the ones started first, or the ones started last, or the ones with the least priority. This decision depends on the algorithm used for killing the process. The fault can be assumed to be a software problem if both the above checks turn nothing out of the ordinary. The ordinary behavior for that node is again determined by its profile and history, obtained from its host base station. A software problem might be one of incompatibility, as, for example, the incompatibilities that exist between the operating system, the windowing wrapper used, the network file system, and the application software [8]. Recovery from such software problems as incompatibilities is possible by notifying the user of the problem, and giving a set of options, such as conversion from one format to another, or continuing in spite of the incompatibility. In the case of SNMP, the MIB (Management Information Base) called HostResources allows the manager to remotely determine what versions of software have been installed.

4.2 A High-Level View of the System

In the following section, we present a high-level view of our system, including the algorithm of its operation.

```
A mobile unit arrives inside a cell and alerts the base unit.
The base unit updates its registers and logs, including time of entry.
The base unit broadcasts to all super agents in that cell.
Super agent in the particular cell acknowledges transmission from the base unit.
The super agent sends a mobile agent to the mobile unit.
The mobile agent copies the parameter values stored in the mobile unit.
The mobile agent reports back to the super agent with the values.
Super agent performs updates to logs and compares values with unit profile.
  if mismatch
    fault detected
    query diagnostic module to determine cause(s) for fault
    lookup table in recovery module for loading procedures onto repair agent
    super agent launches repair agent, which performs recovery at mobile unit
    the repair agent reports back to the super agent
  else
    no fault detected
The super agent makes updates to history module, VLR and HLR.
The super agent kills the mobile agent (and repair agent, if any).
```

5. Conclusion

We have designed an architecture for network management of wireless and mobile networks. It is SNMP-like and is extended by using mobile agents (aglets) to automate the fault management functionality in network management. The system was simulated successfully on UNIX and Windows NT platforms using ASDK from IBM. The system offers attractive features such as low memory usage, fast and efficient information gathering and automated recovery from faults. In addition, the super agent learns from past experiences. After implementation of our system is completed, we plan to test exhaustively for all kinds of typical faults and to evaluate the performance of our system against existing standards.

References

1. Bennington, B.J., Bartel, C.R.: Wireless Andrew: experience building a high speed, campus-wide wireless data network. Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking, IEEE MOBICOM (1997) 55-65
2. Garg, V.K., Smolik, K., Wilkes, J.E.: Applications of Mobile Agent in Wireless/Personal Communications. Prentice Hall PTR, Upper Saddle River NJ (1997)
3. Geier, J.: Wireless Networking Handbook. New Riders Publishing, Indianapolis Indiana (1996)
4. Goodman, D.J.: Wireless Personal Communications Systems. Addison-Wesley Publishing Co., Reading Massachusetts (1997)
5. Magedanz, T., Rothermel, K., Krause, S.: Intelligent Agents: An Emerging Technology for Next Generation Telecommunications? Proceedings of the Fifteenth Annual IEEE INFOCOM (1996) 464-472
6. Rappaport, T.S.: Wireless Communications: Principles and Practice. Prentice Hall PTR, Upper Saddle River NJ (1996)
7. Stallings, W.: SNMP, SNMPv2, and CMIP: The Practical Guide to Network Management Standards. Addison-Wesley Publishing Co., Reading Massachusetts (1993)
8. Zeltserman, D.: A Practical Guide to SNMPv3 and Network Management. Prentice Hall PTR, Upper Saddle River NJ (1999)