Extending SNMP Management to GSM Radio Devices

Sandro Bartolini
Dipartimento Ingegneria dell'Informazione, University of Siena
Via Roma 56, 53100, Siena, Italy

Pierfrancesco Foglia, Cosimo Antonio Prete
Dipartimento di Ingegneria dell'Informazione, University of Pisa
Via Diotisalvi 2, 56100, Pisa, Italy

Abstract

In Telecommunication networks, a high-performance and reliable management of alarm messages is more important than in traditional IP networks. TLC networks comprise up to thousands of heterogeneous Network Elements (NEs) and they generate high alarm traffic due to bad weather, correlated faults and logging activity. In this scenario, alarm messages have an extreme informative value and cannot be lost. TMN framework and CMIP protocol have been proposed to address these problems, but they are very complex to implement, especially on the NE side, where simplicity and low-cost are crucial issues.

SNMP protocol is far easier to implement and SNMP-based management solutions are standard and well established. Unfortunately, SNMP lacks the reliability and performance needed in the telecommunication domain.

In this paper, we present our experience in designing a high performance and reliable protocol over SNMP, suited for the management of a GSM network. We implemented a manager based on such protocol, and integrated it into an SNMP management system.

Siemens ICN has adopted our solution as the reference system for the development and testing of agent/manager SNMP products in the SRAAL family of GSM radio devices.

Keywords: Network management, Telecommunications management, Commercial SNMP management, SNMPv1, reliable alarm delivery, GSM networks.

1 INTRODUCTION

The Telecommunication Management Network (TMN) [1] is a framework for the management of telecom networks [2], while the Simple Network Management Protocol (SNMP) [3] is the most widely used protocol for the management of IP networks and internets [4]. Both the schemas are based on the concepts of *managed objects or NEs*, manager and *Management Information Base* (MIB), though some conceptual differences exist between the two frameworks [3]. Managed elements (e.g. network devices such as hosts, hubs, routers, GSM [12] devices) host a management running process (agent). The manager manages the NEs, exchanging messages with them according to a management protocol. The MIB stores the management information: both the agent and management processes use it to support the storage and the communication of management messages [3].

When managing telecom networks, and especially GSM networks, typical issues are high performance and reliability in the transmission of messages between management station and managed devices. The high performance requirement derives by the high number of devices in the network and their sensitivity to various parameters, such as weather condition, faults, etc. [11]. The reliability issue is due to the need of precise assessment on the quality of the delivered services [6] based on the network status evolution. Besides, a reliable delivery of messages is required to perform alarm correlations, necessary to limit the storm of traps typical of GSM networks [11]. The TMN framework can meet such requirements.

The CMIP OSI protocol [5], on which the TMN framework relies, is too heavy and complex to implement on the agent side. Consequently, device vendors offer ad-hoc management solutions based on custom protocols [3]. The interoperability of these solutions, needed by Telecom Service Provider to achieve an integrated management of the network [3] (usually comprising heterogeneous devices), can be obtained again via a TMN framework and proper adaptors [3], but at an extremely high cost in term of complexity and programmatic effort.

The SNMP protocol, in his SNMPv1 version, requires little computational power on the managed entity. Consequently, most of the IP devices are managed via SNMP. Management solutions based on SNMP are well established, and device vendors offer plug-ins for standard Network Management Systems (NMSs). ISPs and Telecom Service Providers can implement integrated management solutions based on SNMP with a limited effort. However, SNMPv1 protocol lacks reliability in the trap delivery (i.e. an unsolicited message generated by an agent process without a message or event arriving from the manager process [3]). Moreover, SNMP NMSs are designed for traditional IP network and cannot easily handle the alarm traffic of telecom networks. In conclusion, the base SNMP management is not well suited for the telecom domain.

Our idea, in building a new management solution for a network of GSM radio devices, consists of the following steps: 1) adopt the SNMP v1 protocol, 2) solve the reliability issues of the protocol at manager level, and 3) address the performance requirements via an accurate tuning of the implementation. In this way, our solution: 1) can be integrated with standard SNMP NMSs, 2) is able to meet the requirements of GSM networks (we evaluate the performance on a "stressed" environment), 3) can be included with little effort in a TMN framework via a standard Q3-SNMP adaptor [1],[9], as it frees the adaptor from the duty of achieving reliability and performance.

Our implementation tolerates a sustained traffic of more than 10 trap/sec.

The remainder of the paper is organized as follows: Section 2 describes our solution and the proposed protocol, Section 3 describes the implementation issues, and comment the achieved performances. Section 4 concludes the paper.

2 PROPOSED SOLUTION

Application scenario

In this section we describe the origins of the high alarm bandwidth requirements of a typical GSM radio network and highlight the consequent features that a NMS has to meet to properly manage the network.

A GSM network encompasses hundreds to thousands of NEs. Such elements are GSM stations equipped with radio devices that allow the establishment of wireless links between them, and with the corporate network. Radio links are crucial in all cases in which a wired link is unfeasible or too costly (e.g.: not densely populated regions, highly populated city zones not having a usable backbone, next generation systems with very dense access points).

Typically NEs are spread over a very wide area (e.g. cities, regions, states), in the external environment, and rely on wireless links to their neighbors to provide their service. Usually, during bad weather conditions (e.g. storms) NEs can experiment link losses and other malfunctioning [11]. In this situation, NEs generate alarms towards the management infrastructure so that the proper recovery actions can be taken. In addition, upon a NE crash, many stations linked to it can observe a link malfunctioning and can trigger alarm storms to the management system. For these reasons, the management of GSM Networks has to support a bursty high bandwidth alarm traffic: for quite long time periods (i.e. whether time scale) and even higher bandwidth over smaller time periods (i.e. in case of NE faults and alarm chain reaction).

Moreover, NEs usually are programmed to send alarms that report warning conditions in the various components of the managed device. In this way it is possible: 1) to perform logging of service metrics (e.g.: Bit Error Rate in different crucial data-flows and modules); 2) to early detect possible system failures before they occur. Both for logging (monitoring) and alarm correlation (i.e. to highlight the real failure that originated a set of alarms) purposes, it is crucial to avoid alarm losses [11].

These facts highlight that a typical Telecommunication Network requires stronger reliability, and has both the peak and background alarm traffics quite higher than a traditional IP network. Together with Siemens, we have pointed out that a network model generating a sustained alarm traffic of 10 alarms/sec can effectively represent the depicted scenario.

Architecture of the solution and Integration Strategy

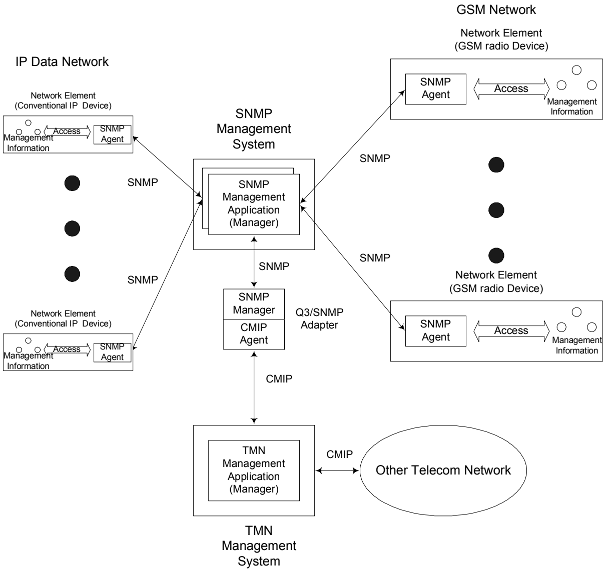


Figure 1: Architecture of the proposed solution and integration with other telecommunication and data networks.

Our proposed solution to manage a network of GSM devices is depicted in Figure 1. Each GSM device implements an SNMP v1 agent (i.e. a software process which communicate with the manager via the SNMP v1 protocol on top of the UDP/IP stack) which accesses management information, such as the equipment type, the location, the IP address, the software version, the status of the device, etc. The agent exchanges management information with a manager process (SNMP Management Application in Figure 1), which is part of a SNMP Management System. In particular, the agent replies to the manager request, and reliably sends traps when the state of the device changes. To achieve reliability, we develop and implement an application protocol on top of SNMP v1, based on the acknowledgement of received traps, trap time-stamping and time-outs. Details of such protocol are given in the next sub-section.

In a typical TSP (Telecommunication Service Provider) or ISP (Internet Service Provider) deployment, the SNMP Management System is used to manage also other devices, typically the ones belonging to IP Data Networks, i.e. routers, switches, printers, hosts (left side of Figure1, where they are managed by the other SNMP Management Applications). The management of such devices is well established, and, the SNMP Management System, in Figure 1, is typically a commercial available product. Our management application is required to interoperate with this product, while achieving the performance need of a GSM network. We describe in Section 3 the implementation issues deriving from such an assumption.

Finally, the GSM network, as part of a Telecommunication Network, can be integrated in a TMN framework. This can be done via a Q3-SNMP adapter: it is a software component which transparently translates the TMN CMIP management protocol primitives in SNMP messages. The Q3-SNMP adapter is a mature and commercial available product; details on it may be found in [9].

Details of the communication protocol

We implemented a reliable protocol over SNMPv1 in order to meet the reliability requirements of alarms originating from NEs. In the following, we describe the details of the protocol; in section 3 we discuss the implementation issues of the protocol and the achieved performance in managing devices.

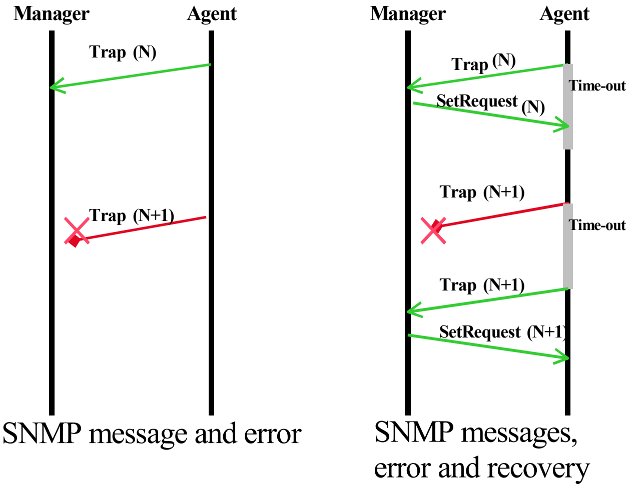
We choose SNMPv1 as the base protocol because, though it provides unconfirmed delivery of alarms while SNMPv2 and v3 have a confirmed alarm delivery system (Inform Message) [3], the SNMPv1 is the de-facto standard in the industrial implementation of SNMP-agents [4]. As a consequence, most of the implemented agent are SNMPv1 compliant and SNMP v1 and the other SNMP protocols are not interoperable (SNMP v1 entities cannot directly exchange messages with SNMP v2 and v3 entities [8]). Thus, implementing a SNMP management solution with protocol versions other than v1, prevents the interoperability of such a solution with most of the already developed agents.

Figure 2 highlights the kernel features of our protocol, that it is based on time-stamping (numbering) of SNMP traps. The agent, after sending a trap to the manager, starts waiting for the acknowledgment of the trap. Upon acknowledgment arrival, both communication partners (management application and NE) are sure of the trap delivery. Timeout and retransmission mechanisms, together with the time-stamping, allow to detect alarm-message losses (fault detection) and trigger the line-up of the management station with the NE alarm status (fault recovery).

The protocol implementation requires the agent to implement a MIB variable *trapNum* that holds the progressive trap timestamp. After a trap-send and the corresponding ack-reception, the agent increments the *trapNum* counter. If the trap or the ack get lost, the agent's timeout (T1 seconds) fires and causes a re-transmission of the same trap (with the same *trapNum*). The agent is programmed to perform N1 trials before going into *idle* mode, where it does not consume bandwidth, and waits to be waked up again by the manager.

The manager has to maintain the last acknowledged trap's timestamp for each agent. In this way, when it receives a trap, it can check if there have been losses in the meantime. If the new trap's timestamp is in sequence, the manager acks the trap through a SNMP-Set on the agent's appropriate MIB variable. Otherwise, the manager: 1) starts the line-up process, in order to retrieve the up-to-date agent status; 2) turns the agent into *active* mode (i.e. enables him to send traps again), as the agent can go into *idle* mode during the line-up activity.

The acknowledgments are performed on a per-trap basis. Obviously, some optimization to this scheme may be introduced (essentially, a go-back-N scheme [7]), if the base protocol's performances are not enough for the specific application domain.



SNMP message and error

SNMP messages, error and recovery

Figure 2: (left) SNMPv1 trap message transmission. An error is not recovered. (right) In our protocol each trap is numbered and acknowledged by a SetRequest message. This guarantees message loss detection and recovery.

Transient behaviors: our protocol has to cope with different faults either on the agent side, communication or manager side. Here we describe how the protocol works in some significant fault conditions that may occur.

If the agent crashes, the manager cannot rely on a trap informing of this fact from the agent. Therefore it has to poll the agent for its aliveness, with a properly tuned poll-frequency. Therefore, trap management and poll activity have to be integrated together in the manager.

When an agent comes up (e.g. a new device is added to the network, or it is switched-on after a repair activity) it is in *idle* mode, and the manager detects it through polling. Then, it lines-up with the agent's status (i.e.: reads the internal device status and *trapNum* value) and sets the agent into *active* mode.

If the communication between agent and manager crashes during an ack transmission, the agent sends again the same trap to the manager. The latter understands from the timestamp that it is a copy of the previous trap and performs the ack again without doing anything else, as all the actions for that trap (e.g. status update, logging) have already been done.

Finally, if the manager crashes (e.g. management software crashes, it is switched-off) all the agents in the network observe, after an assigned time-out, a missing ack upon their trap messages. In this way, they eventually stop sending repeated traps and put themselves into *idle* mode.

3 IMPLEMENTATION ISSUES AND PERFORMANCE EVALUATION

As stated in the previous section, the Network Management System of Figure 1 is typically a commercially available product. We describe in this section how it influences and helps the implementation of the trap acknowledgement application and the achieved performance.

The trap acknowledgement mechanism, in NMSs, can be implemented in two ways: through automatic actions [10], i.e. processes triggered by the NMS on the occurrence of configured events, or through a daemon, which reliably receives events by the main message manager (mmm). The mmm is a standard component of a NMS. Other components (in Figure 3) are the snmp receiver, which listens for incoming SNMP messages and forwards them to the mmm; the Active Alarm List, which maintains the current state of the network and displays it on a GUI; a database stores all the received event (the platform log). All NMS processes can be configured through specific configuration files. The communication between the daemon and NE may be implemented via the low level (C or C++) API offered with the NMS.

In the case of automatic actions, (Figure 3), traps arrive as UDP messages, they go through the SNMP trap receiver and the main message manager (mmm), which triggers the required action (i.e. a process executing a utility program or a script). This process acknowledges the NE trap through a SNMP-set message. Each trap causes a new process spawn: such overhead may lead to unacceptable performance even on a high performance machine.

In the second case, a daemon is permanently up and linked to the main message manager so that it can receive reliable notification of trap arrivals. Upon each notification, the daemon acknowledges the traps, according to the protocol described in the previous session.

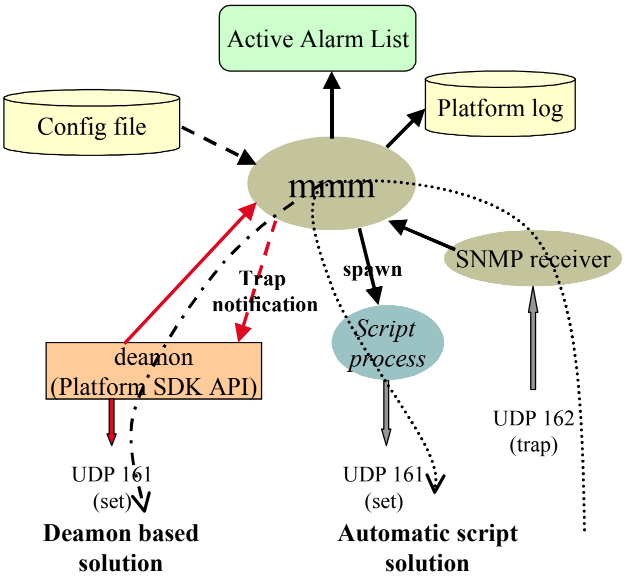


Figure 3: Typical components of a Network Management System and detail of the trap flow from the NE to the manager, its acknowledgement (SNMP Set) and logging. On the right, the dotted line highlights the automatic action based solution, where a process is spawn for each trap by the Network Management System. In the daemon based solution, the main message manager (mmm) reliably notifies (dot-segment line) the daemon to perform the acknowledgement through API calls.

The advantage of the first solution relies on its simplicity: it can be implemented with little effort and by utilizing standard features available with NMSs [10]. However, the continuous spawning of processes, when the system is managing a network of GSM device, where there is a lot of alarm traffic between manager and agents, can lead to unacceptable performance degradation and can stress too much the processes creation mechanism of the underlying operating system.

On the contrary, the second solution requires no spawning of processes, because the daemon is always up. Besides, as the communication between mmm and daemon has been implemented via low level API, it is more performing than the first one. The disadvantage of such a solution is that it requires programmatic effort, due to the distributed nature of the problem.

We implemented both the solutions by adopting a commercial available NMS running on a WindowsNT/2000 operating system. We compare their performance in a "stress scenario" to assess the solution's ability to achieve the required level of performance (an eligible solution must be able to manage at least 10 trap/sec in the steady state).

In our "stress scenario", each GSM device is simulated via a specific Siemens software that implements the agent part of the protocol, and generates trap sequences typical of a real environment. This network simulator operates in a “steady fault condition”: it generates a new trap as soon as the manager has acknowledged the previous trap. The simulated network consists of 500 NEs, connected on a 10Mbit LAN. The manager runs on a Windows 2000 Server, hosted by a 2GHz Pentium IV.

We measured the Trap Response Time, i.e. the Time between a trap transmission and the ack to the related Set performed by the manager. Such a time must be less than 100 msec, to guarantee a steady rate of 10 trap/sec. As Figure 4 shows, only the daemon solution is able to achieve, in our environment, a trap response time that is far below the spec requirements.

These results show that, with a careful implementation, the SNMP v1 protocol can be utilized to manage a network of GSM devices. However, standard tools available in SNMP NMS (such as the automatic actions), are not suited to implement such solutions, as they are designed for IP data networks, which exhibit less management traffic than GSM networks.

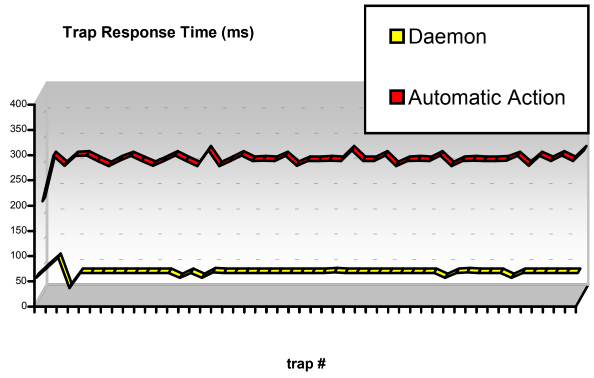


Figure 4: Time necessary to process a trap (Trap Response Time) vs. trap number. Only the “daemon” implementation is able to satisfy the design requirement of the system (response time less than 100 msec.)

4 CONCLUSIONS

With this paper, we report our experience in implementing a network management solution, which integrates the management of a network of GSM Radio devices in a SNMP framework.

SNMP based management can allow a standard approach to the management of the heterogeneous devices that compose a telecom network. Moreover, the adoption of the SNMP protocol allows easier implementation of agents and MIBs if compared to other approaches like the TMN/CMIP framework.

We extended the SNMPv1 protocol to guarantee no loss of trap messages, and implemented it via a dedicated deamon-manager to meet the performance requirement of a GSM network.

Siemens ICN has adopted our solution as the reference system for the development and testing of agent/manager SNMP products for the SRAAL family of GSM Radio Devices.

5 ACKNOWLEDGEMENTS

The present work has been carried out in a framework of collaboration between the Dipartimento di Ingegneria dell’Informazione, University of Pisa, Pisa, Italy, AGF S.a.S., Milano, Italy, and Siemens ICN, Cassina dei Pecchi, Italy, for the development of a protocol for reliable communication between manager and agent in GSM radio networks. We are grateful to Massimo Bernini and Alberto Grazioli (Siemens ICN), which gave us the necessary support and data. Francesco Molè helped us in the starting phase of the project. Ruggero Calò and Simone Baldi contributed to the development and performance evaluation phases.

6 REFERENCES

[1] ITU-T Recommendation M. 3010, Principles for a Telecommunications Management Network, July 1996.

[2] D. J. Sidor, TMN Standards: Satisfying Today’s Need While Preparing for Tomorrow, IEEE Communications Magazine, Vol. 36, N.3, pp. 54-64, March 1998.

[3] M. Subramanian, Network Management: principles and Practice, Addison Wesley, Reading, MA, 2000.

[4] W. Stalling, SNMP and SNMPv2: The Infrastructure for Network Management, IEEE Communications Magazine, Vol. 36, N.3, pp. 37-43, March 1998.

[5] Y.Yemini, A Critical Survey of Network Management Protocol Standards, Telecommunication Network Management into the 21st Century, IEEE press, Chapter 2, pp. 19-71, 1994.

[6] K.H.Muralidhar, Knowledge-based Network Management, Telecommunication Network Management into the 21st Century, IEEE press, Chapter 7, pp. 200-233, 1994.

[7] A. S. Tanenbaum. Computer Networks - Third Edition. Prentice-Hall International, Inc., 1996.

[8] D. Perkins, E. McGinnis, Understanding SNMP MIBs. Prentice-Hall, Inc., NJ, 1997.

[9] A. Keller, Tool-based Implementation of a Q-Adapter Function for the seamless Integration of SNMP-managed Devices in TMN. NOMS98 IEEE/IFIP Network Operations and Management Symposium, Vo. 2, 15-20 Feb 1998, pp. 400 –411.

[10] W. Stalling, SNMP, SNMPv2, SNMPv3, and RMON 1 and 2, 3rd Edition. Addison-Wesley, NJ, 1999.

[11] H. Wietgrefe, Investigation and Practical Assessment of Alarm Correlation Methods for the Use in GSM Access Networks. *NOMS 2002* - IEEE/IFIP Network Operations and Management Symposium, no.1, pp.391-404, April 2002.

[12] European Telecomm. Standards Institute: GSM system Standard Series.