Submitted to the International Conference on Information Networks and Data Communication, INDC 94, Funchal, Madeira Island, Portugal, April, 18-21, 1994.

A Monitoring System for Local Area Networks Using Distributed Measurements

W. Schollenberger

Institute of Communications Switching and Data Technics, University of Stuttgart,
Seidenstraße 36, D-70174 Stuttgart, Germany

Abstract

Network monitoring is an important aspect of network management. Due to the variety of communication protocols in today's Local Area Network environments, the demand emerges to monitor the network on the Data Link Layer. On this layer, the network is divided into subnetworks, thus requiring the monitoring to be distributed to obtain a global view. Management platforms concentrate on network nodes and lack of convenient methods to monitor the network itself. They do not adequately consider the unique characteristics of each network and the different demands of network managers. This paper discusses various aspects of network monitoring and introduces distributed measurements as a flexible way to monitor a network. Based on this concept, a system is presented with an open architecture, providing for new measurements and respective equipment to be easily inserted.

Keyword Codes: C.2.5; K.6.4; 3.2
Keywords: Local Networks; System Management; Management Tools

1. INTRODUCTION

During the last few years, Local Area Networks (LANs) developed from small, stand alone installations with simple topology to widespread, hierarchically structured networks, interconnected by means of repeaters, bridges and routers. More and more, network management turns out to be a difficult and complex task. In this context, network monitoring is necessary to provide an actual view of the network components' status and their traffic load. It is important to recognize critical states, such as overload situations and broadcast storms in order to react quickly.

Since internationally standardized protocols as the OSI seven layered protocol suite are far from being used exclusively, today's networks are cluttered up with different protocols for workstations, PC-networks and computer clusters. These protocols share the same medium and their traffic characteristics influence each other. Therefore, it is necessary to monitor a network on a layer common to all protocols, such as the Data Link Layer or Media Access

Control (MAC) Layer in Local Area Networks. At this layer, however, a network is divided into independent subnetworks, each having its own traffic characteristics. A packet is not present on all subnetworks, and collision rates (at networks using ISO 8802/3 MAC protocol [1]) are varying on different segments. In order to get an image of the entire network, monitoring has to be performed in every subnetwork by distributing some kind of sensors and by concentrating the results at a central site.

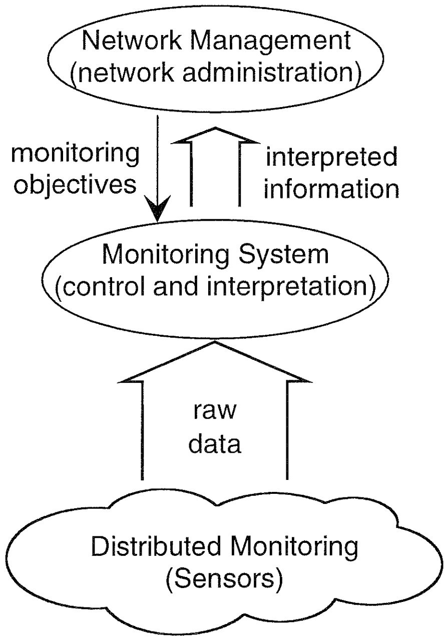

Figure 1. Flow of monitoring information

As depicted in figure 1, distributed monitoring produces a large amount of raw data which normally outstrips a human network administrator or network management applications [2]. A monitoring system filters and interprets the collected data to reduce quantity and to increase informational substance. Since the objectives of network management vary strongly, the preprocessing functions have to be adaptive.

This paper discusses aspects of network monitoring and different ways to perform it in a distributed manner. It is shown that the concept of distributed measurements leads to the flexibility a network administrator needs to tailor monitoring to his network. A monitoring system is presented that realizes this measurement concept in its architecture and controls a distributed measurement system. It is designed as an open system that allows the introduction of new measurements and measurement equipment and that serves the needs of different users by providing an adequate presentation of the measurement results. The monitoring system is not restricted to LANs. It can also be used in Telecommunications Management Networks (TMN) and other areas of distributed monitoring.

2. NETWORK MONITORING AND NETWORK MANAGEMENT

Network management has been the subject of various standardization efforts. The Simple Network Management Protocol (SNMP) as a part of the TCP/IP protocol suite is a practical approach while OSI network management is more complex, powerful and flexible. Both approaches share the definition of two different types of management processes. They are designated as *manager* and *agent* and have a client-server relationship. The duty of the agent process is to obtain information about network resources and present it to the manager without interpretation, organized as a conceptual database called the Management Information Base (MIB). The Remote Monitoring Management Information Base (RMON-MIB) [3] for SNMP and the OSI standard that definines the Workload Monitoring Function [4] provide frameworks how to place data related to network monitoring into the MIB. Agents act as sensors while it is up to the manager to collect and evaluate the information. Little is said about how this should be done and which data should be retrieved.

It is widely accepted to divide the management functions into five groups called Systems Management Functional Areas [5] which cover the whole life cycle of a network from the planning phase to installation, operation and reconfiguration. Each area has its own monitoring aspects:

* *Configuration Management* starts with planning a network and mainly deals with the logical and physical topology, with network names, addresses and protocol parameters. In the operational phase, network monitoring is performed to obtain the status of the network's nodes and the network itself. The configuration model is compared to reality by checking network addresses and routing information in the nodes and by examining the address information in packet headers in order to discover new stations.

* *Performance Management:* Network performance has to be evaluated, supervised and predicted. Evaluation of network performance implies monitoring of data throughput, error rates and network load. Protocol usage statistics and traffic matrices assist in getting more detailed information about the behaviour of network nodes [6]. Long time statistics result in important input to configuration management for reconfiguration decisions.

* *Fault Management:* It is important to achieve a low delay between the occurrence and the detection of a network fault. If there is a problem at a particular node and communication is still possible, the network management agent can inform the manager station by means of event reports or traps. If not, other nodes will have to detect that the respective node is unreachable. By periodically monitoring the connectivity among all stations in the network, the network administrator is given the chance to discover that a node is unreachable. Problems with the network media are mostly caused by lose connectors or physical damage to a cable. This kind of problems doesn't necessarily lead to a complete network failure. Sometimes, only high collision or error rates are identified. Network traffic monitoring shows the effects of transmission problems, while monitoring the medium itself quickly points out the reason and location of the fault. Special hardware is required to observe the medium at this level. If physical problems are likely to occur, this equipment should be permanently installed and remotely controlled (e.g. in a manufacturing environment). Faults occurring sporadically are difficult to trace because they disappear before they can be

located. No general algorithms exist to detect and locate network faults. In practise, however, strategies for fault analysis emerge from the experience gained by everyday network operation.

* *Security Management:* The primary aspects of Security Management are to ensure that confidential data on the network can only be interpreted by the intended receiver and to prevent entities from using network resources without authorization. This is mainly achieved by data encryption and authentication mechanisms. Security Management describes how these mechanisms are applied and provides functions to detect, log and avert security attacks [7]. These functions can be performed by nodes belonging to a secured connection only since they possess the appropriate encryption keys to decode the messages. In a network where secured communication is applied, it is difficult to use protocol analysers to solve networking problems.

* *Accounting Management* records the usage of network resources and relates it to the corresponding user. A possible way to account network traffic consists in recording all packet headers and then reading source and destination addresses in order to set up a traffic matrix comprising all stations. Problems occur if routers are involved since they replace the MAC-addresses stored in the packet header. Tracing a packet over several subnetworks involves decoding and interpreting the address information added by the network layer. This layer, however, varies among different protocol suites. A better approach is to charge the usage of remote networks only and to record accounting information at the interconnection device (router, gateway). If the coupling device lacks of this functionality, it is still possible to monitor the traffic to and from that device.

3. DISTRIBUTED MONITORING

In general, distributed monitoring is performed by sensors (probes) distributed in the network and by a central station. The sensors gather network information at the source while the central station collects and interprets the data from the sensors and presents the results to the user. Approaches to distributed monitoring can be classified regarding the amount of preprocessing that is performed by the sensors. The most simple and universal way is to collect the local information and send it to the central station without reduction or analysis within short intervals. All information is available at the central station and, as a consequence of this, no restrictions on monitoring functions are imposed. The sensor needs no storage capacity and little processing power. For this reason, the sensors are simple and cheap, so many of them can be installed at reasonable costs. The reception of data from the sensors implies proper connectivity to the respective subnetwork. If, on the other hand, a large amount of information is recorded or many sensors are installed, the network load originating from monitoring is considerable. [8] and [9] describe two examples where this approach has been realized.

To reduce network load, information retrieved by the sensors can be preprocessed, e.g. by calculating statistical summaries. The results are transmitted in a periodic but less frequent way. Monitoring functions are restricted to the sensor's abilities. Therefore, the central station should be able to adapt the preprocessing functions to the needs of the network administrator.

Sensors of this kind need more processing power but still no storage capacity. A protocol has to be implemented to control the sensor.

The third class of sensors analyses the local information completely and enables the central station to retrieve the desired results. This concept is realized within the SNMP RMON MIB and some commercially available network monitors [10], [11]. History logs are kept within the sensor and alarm reports are sent to the central station if some monitored value reaches a defined threshold. Only information required by the central station is transmitted. Due to the lack of periodic transmissions, however, no statement can be made about the connectivity to the respective subnetwork. Sensors of this kind require more memory, are more complex and consequently more expensive.

In addition to the classes described above, ordinary nodes can be considered as sensors since they hold monitoring related information that can be accessed via network management protocols.

Two possible communication paths between the central station and sensors can be distinguished:

* *Inband communication* employs the network that is subject of monitoring. No additional installation is needed, but if the entire network fails the monitoring equipment is useless. Fortunately, total network breakdowns are rare. Successful communication implicitly assures the connectivity to the appropriate network segment.

* *Outband communication* uses a different, separately installed medium. This amount of security causes additional costs.

It is reasonable to use inband communication while the network is operable. During network failures, dialup modem connections can be employed to realize outband communication.

Monitoring tasks usually arise during network operation and are difficult to predict. Monitoring activities can be divided into two general classes:

* *long term monitoring:* Network load, user activities, unknown addresses and similar data is recorded for several weeks or months. The measurement interval is adjusted to suit statistic purposes and is far too long to react on faults in real time. Information retrieved in this fashion is stored to an archive.

* *short term monitoring:* Information is collected to discover malfunctions and to obtain an actual image of the network. To perform short term monitoring, a huge amount of data has to be processed in real time. In case of network problems, the network administrator is informed by alarm reports which may start automatic fault diagnosis procedures [12]. Logs that record recent network activities are useful to analyse problems after their occurrence.

4. MEASUREMENTS IN LOCAL AREA NETWORKS

By means of network monitoring, a lot of different topics can be observed, each of them being useful in some case. In general, however, only few network parameters are recommended to be monitored, e.g. the status of important nodes, simple network load or error rates. The administrator has to define the correct monitoring level for a given network. Therefore, a monitoring system must provide the flexibility to define the scope of interest. It must be considered that monitoring equipment from different vendors has to be integrated into the system.

In typical network monitoring, a number of independent activities are performed in parallel. Any activity can be treated as a single measurement task. The sensors are distributed all over the network and form a permanently installed measurement system. The monitoring system supervises all measurement tasks and calculates the results.

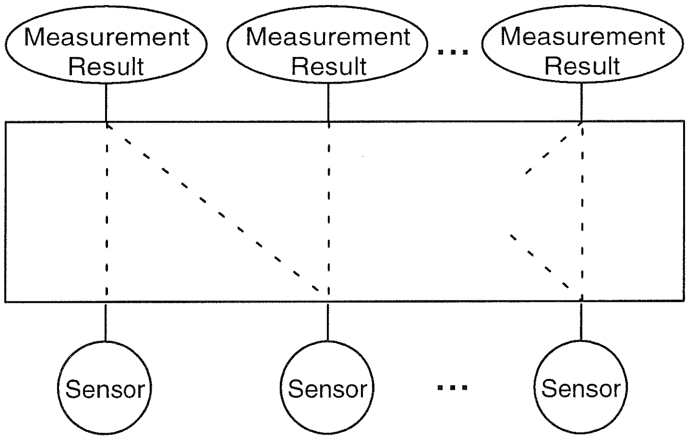

Figure 2. Relation between measurement results and sensors

Figure 2 shows how data collected by the sensors is processed to obtain measurement results. In general, several sensors cooperate to perform a particular measurement function. However, every single sensor can be used in many different measurement contexts. The flexibility of this method arises from the variety of possible combinations of sensors, measurement functions and result evaluation procedures. Usually, there is a discrepancy between a measurement request defined by a user and the capabilities of the measurement equipment. The central station performs user oriented measurements by means of base measurements that are executed by the sensors.

5. CONTROLLING DISTRIBUTED MEASUREMENTS WITH A CENTRAL MONITORING SYSTEM

5.1. Overview

At the Institute of Communications Switching and Data Technics at the University of Stuttgart, a distributed measurement system for networks according to the ISO 8802/3 (Ethernet) standard has been developed in a cooperation project with Siemens, Germany. It consists of measurement stations, each containing a processor board with network access facilities and measurement modules for the physical layer, e.g. an on-line time domain reflectometer board.

For further detail, please refer to [13]. Figure 3 shows the measurement stations with their software components, realized as processes under a simple operating system. Every measurement application is associated with a measurement module. A measurement application receives base measurement requests, executes them by controlling the measurement modules and generates the base measurement results. The communication subsystem is responsible for data transfers between the measurement application and the central monitoring system. Additionally, it responds to LLC (Logical Link Control) test frames. A separate measurement application creates test frames in order to check for the connectivity to other measurement stations.

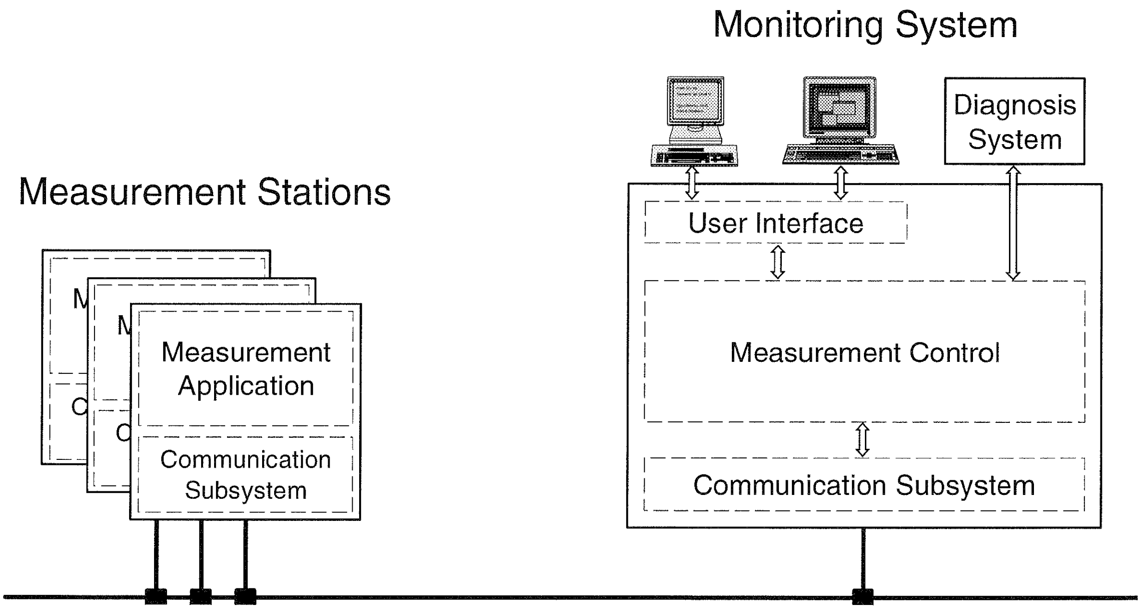

Figure 3. Monitoring system with distributed measurement system

All measurement functions based on the analysis of packets on the medium can be realized on the processor board. For the execution of these functions, the LAN coprocessor is set to a promiscuous mode to receive all packets [14]. Implementation has shown that a small amount of hardware (Intel 80186 CPU with 82586 LAN coprocessor) is sufficient to record network load and network errors and to set up a traffic matrix for packets, bytes and errors. Since these boards are distributed all over the network, simple LAN analyser functionality is available at every network segment.

The monitoring system accepts user requests, supervises the measurement process, interprets the sensor data and presents the results. The monitoring system performs a user defined measurement by dividing it into one or more base measurements. The measurement control processes the base results and presents them to the user as a single measurement result. It keeps track of all measurements in the system and must be extremely stable in order to prevent long time measurements from being disrupted.

There are three types of users which have to be served by the user interface. The network administrator performs several standard monitoring tasks in parallel and has to be aware of the status of the entire network at a glance. To ease comprehension, measurement results should be presented in a graphical form. The second type of user is the network specialist who tries to trap network faults with the monitoring system. It is likely that he controls the monitoring system via a remote, alphanumerical terminal. All features of the monitoring system can be taken advantage of by a command line interface. An expert system for fault diagnosis in LANs [12] represents the third type of user. During a diagnosis session, automatic symptom test methods start measurements without user intervention. The user interface module adapts both the command line and the graphical user interface to a common interface which is also used by the diagnosis system.

5.2. The Architecture of the Monitoring System

The monitoring system should be designed as an open and extendable system in order to allow other remotely controllable measurement equipment to be integrated. Along with the demand of high stability, this suggests an architecture of cooperating processes running on a UNIX platform. A measurement request is handled by an independent measurement process executing a measurement program. To extend the monitoring system to support new measurements or new components, new measurement routines can be developed and easily integrated into the system. The generation of measurement requests (jobs) and the evaluation of measurement results are realized as processes and separated from the measurement process. They communicate mainly via shared memory elements where all important information about the jobs in the system resides. Since shared memory elements can be accessed by other processes, a maintenance process can remove a job completely without affecting other measurements in case a measurement request cannot be completed for some reason. This is accomplished by terminating the respective processes and by cleaning up the corresponding memory structures. The system's flexibility is achieved by the possibility to combine different measurement programs and result evaluation routines for a particular measurement request.

The internal architecture of the measurement control is depicted in figure 4. The communication subsystem and the user interface dispose of well defined interfaces. The job list is the key element and grants access to all measurements. It is realized as a shared memory element and provides the communication area for all processes involved in one measurement. Concurrent access is resolved by means of semaphores. The job list is preceded by a header that contains the number of jobs in different states (free/ready to run/processing/finished). Each measurement occupies an element in the job list which contains the job's state, the measurement type, the starting time and the shared memory identifier of the job element. During the execution of the measurement, the identifiers of the measurement process and the job evaluation process (PID's) are stored in the job list element.

The job creation process generates a new measurement by occupying an empty job list element and by allocating a new job element to hold the parameters of the associated measurement. In this job element, the results of the measurement process are stored. The job evaluation process reads the measurement results from the job element and presents them to the user. Concurrent write access cannot occur, as only one process is permitted to write to a job element at every stage of the measurement. During periodic measurements, synchronization between the measurement process and the job evaluation is achieved by transmitting intermediate results via a message queue.

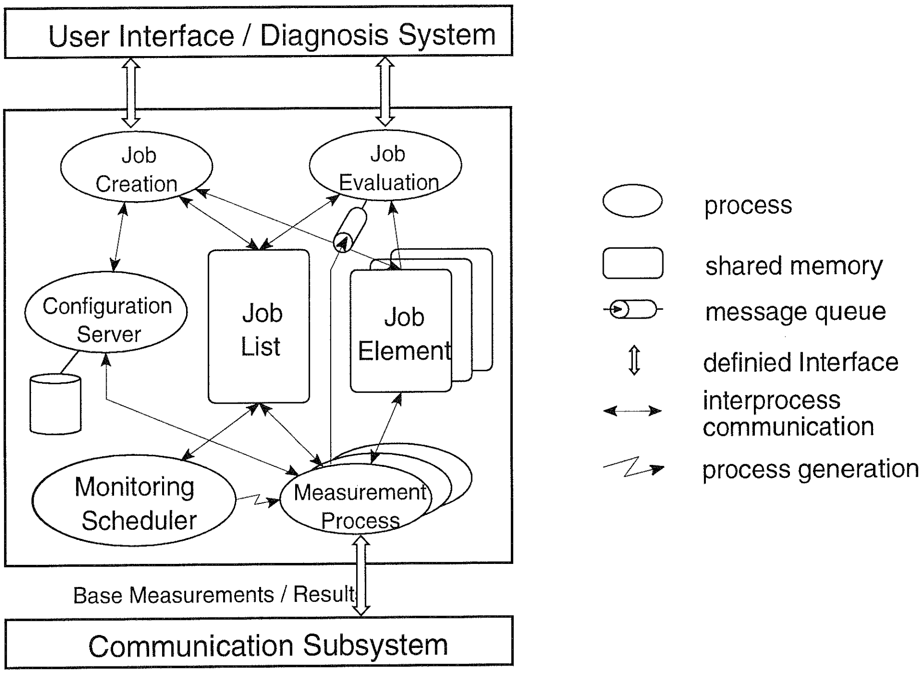

Figure 4. Architecture of the monitoring system

5.3. Processes of the Measurement Control

5.3.1. The Job Creation

A measurement request description language has been defined to specify the measurement type, the measurement parameters and the processes involved. While some parameters are common to all measurement functions, others are specific to a particular class. The parameters describing the temporal behaviour of a measurement are explained by means of a model depicted in figure 5.

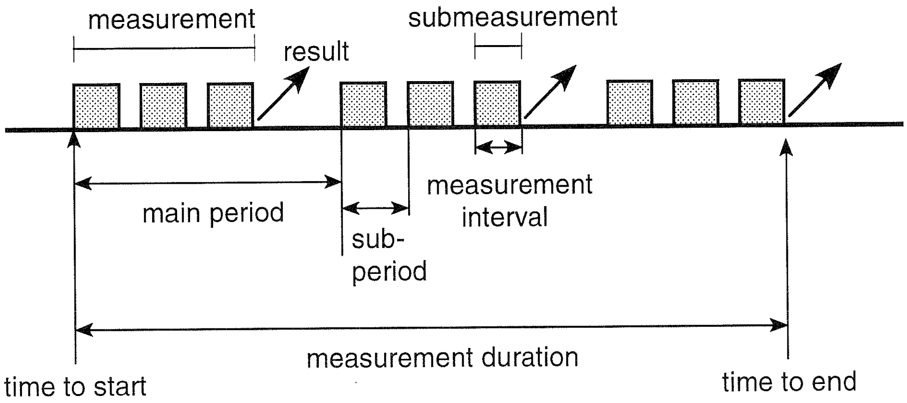

Figure 5. Model for the temporal parameters of a measurement request

The starting point of a measurement can either be defined absolutely or relatively to the current time. Periodic measurements are restarted at the end of the main period. If a measurement is composed of different submeasurements, the result serves as a statistical summary. Another model has been defined for connectivity tests and related measurements. Primary stations are controlled by the monitoring system and perform the requested task by sending test frames to a set of secondary stations.

The job creation process interprets character strings according to the measurement request description language in order to determine the measurement type and the corresponding parameters. Address parameters can be given as symbolic names or network addresses and are resolved by the configuration server. A newly created measurement is represented by an entry in the job list and a job element. The measurement is started, when the monitoring scheduler creates the corresponding process.

5.3.2. The Monitoring Scheduler

The monitoring scheduler is realized as a permanent background process. It sleeps until the job creation indicates that a new measurement has been generated. The job list is scanned to locate the corresponding element. The monitoring scheduler determines whether the measurement process has to be started at once. In this case, it creates a measurement process by invoking the measurement program specified in the job list element. If the measurement is delayed, a timer is set and the scheduler waits for the corresponding timeout or for a new measurement request.

The parameter *output mode* is stored in the job list element and controls whether the job evaluation process is started in conjunction with the start or the termination of the measurement process ("at once"/"when done"). As an option, the evaluation process can be initiated by the user ("by shell"). For simple measurements, the output mode is set to "when done" or "by shell". Intermediate results during periodic measurements can be obtained by starting the job evaluation "at once".

5.3.3. The Measurement Process

The measurement process divides the measurement into base measurements and tries to reserve the participating measurement stations. The measurement will be aborted if the reservation continues to fail. The job creation assures that reservation conflicts can be resolved by slightly delaying one of the involved base measurements. Results are obtained by processing the data originating from these base measurements. They are written into the respective job element. The result mode parameter in the job list element defines the way how measurement results are stored in the job element (overwrite/append) and whether intermediate results are to be sent to the message queue of the job evaluation process. The queue can hold a certain number of messages, so no measurement result is lost because of processing delays in the job evaluation.

5.3.4. The Job Evaluation

The job evaluation process prepares the measurement results for display and transmits them to the user interface or the diagnosis system. Any process can access measurement results since they reside in shared memory. Intermediate results that are transmitted to a message queue can only be read by the intended destination process. The communication with the user interface is message based. The job evaluation process creates output windows by establishing connections to the user interface. The type and format of messages that are sent to the user interface depends on the corresponding display context. Examples for different display contexts are text windows and graphical windows displaying diagrams or network maps.

5.3.5. The Configuration Server

The configuration server holds configuration data concerning the measurement system and the network itself. Implemented as a process, it receives requests via a message queue and sends the corresponding replies back to the requesting processes. Concurrent access to the configuration server is resolved by serializing the competing requests. Due to this realization, the internal representation of the database is hidden and can thus vary from simple text files to the usage of an SQL database or even the Directory Service [15]. The present implementation is based on a relational data model and is realized by using a SQL database system. In order to increase the performance, information being frequently accessed or modified is written to files that serve as a kind of caching mechanism.

The configuration server maintains the occupation status of all stations that are able to deliver some kind of measurement data. Furthermore, it provides a reservation service that is used to prepare a measurement by occupying the participating stations.

5.4. Implementation Details

The monitoring system has been implemented on a PC386 under SCO-UNIX (Open Desktop) using the C programming language. Network access is provided by the Berkley layer-2 streams interface. The graphical user interface is based on the OSF/MOTIF environment. Software for the measurement systems has been developed using the IC86-compiler from Intel. This compiler is able to generate code that can directly be used for EPROM devices. Basic multitasking capabilities are provided by the Queue Operating System (QOS) that has been development at the institute.

6. CONCLUSION

Information provided by network nodes and special measurement equipment has to be collected and adequately processed in order to monitor a network in its entirety. Any monitoring activity can be considered as a separate measurement task. This paper introduces a monitoring system that meets this idea by providing for the definition and the execution of complex measurement requests. The diversity of today's heterogeneous measurement environments is hidden by an integrated user interface. Since the monitoring system features an open architecture, it can be easily expanded and adapted to newly arising demands.

References

1. ISO/IEEE 8802 Part 3: “Information Processing Systems - Local Area Networks - Carrier Sense Multiple Access with Collision Detection”.

2. Brusil, P.; Stokesberry D.; Daniel P.: "Towards a unified theory of managing large networks", IEEE Spectrum, no. 4, pp. 39-42, April 1989.

3. Waldbussen, S.: “Remote Network Monitoring Management Information Base”, Internet Draft, July 1991.

4. ISO/IEC DIS 10164-11: “Information Processing Systems - Open Systems Interconnection - Systems Management Part 11: Workload Monitoring Function”, 1992.

5. ISO IS 7498-4: “Information Processing Systems - Open Systems Interconnection - Basic Reference model - Management Framework”, 1989.

6. Mogul, Jeffrey C.: Efficient Use of Workstations for Passive Monitoring of Local Area Networks, Communications of the ACM 33 (9), pp. 253-263, 1990.

7. Garbe, K.: “Sicherheitsstandards für offene Kommunikationssysteme” (German), PIK 13, pp. 139-145, 1990.

8. Jander, M.: “Putting Network Monitoring at EASE”, Data Communications, March 21, p. 49, 1993.

9. "Ethernet box, Benutzerhandbuch" (German), RzK Doris Köpke, Asbach Ww., 1992.

10. “HP LanProbe System User/Reference Manual”, Hewlett Packard Company, Palo Alto, 1992.

11. Distributed Sniffer Product Sheet, Network General Corporation, Menlo Park.

12. Schröder, J. M., Schödl, W.: “A Modular, Distributed Knowledge Base for Local Area Network Diagnosis”, Second International Symposium on Network Management, Washington D.C., April 1991.

13. Schröder, J. M.: “Monitoring und Diagnose in lokalen Netzen” (German), Elektronik 40. Jg., issue 7, pp. 158-164, issue 9, pp. 76-80, 1991.

14. Ball, E.; Protogeros, A.: Traffic analyser and generator”, Computer Communications Vol. 13 (7, 8), pp. 407-413, 469-477, September 1990.

15. CCITT Recommendation X.500 The Directory - Overview of Concepts, Models and Services, December 1988.