The Simulator of SHR Telecommunications Network*

Wen-Chien Liu[1], Ge-Lian Chao[1], Yeong-Shyan Sun[2], and Chyan-Goei Chung[1]
[1]Institute of Computer Science and Information Engineering,
National Chiao Tung University, Hsin-Chu 30050, Taiwan
[2] Chunghwa Telecommunication Laboratories, Chunghwa Telecom Co., Ltd, Tao-Yuan, Taiwan
{wcliu, glchao, yssun, cgchung}@csie.nctu.edu.tw

Abstract

The SHR network is the telecommunications network of new generation with the high bandwidth and reliability. Along with the new network, an associated telecommunications network management system (TNMS) must also be built, cooperating with the OAM functions of network to completely ensure the reliability and high-quality of services. To test the functionality of TNMS, a testing environment of SHR network is required to provide the OAM operations and the associated network operations, including the normal and the abnormal (such as the protection switch on failure or shutdown of network equipment.) The testing environment is better to use a simulator of network rather than a physical one, where the operations of network are easier to control, observe, thus test. Besides, when the network is under construction, the fault and configuration management operations can only be provided by the simulator but for testing. Therefore, in this paper, we design a SHR network simulator for the testing of TNMS. The simulator is composed of 1) NESSs each of which simulates a central office and its associated network elements (NEs) and agents, 2) a herald which responds to exchange the management information with TNMS, and 3) a network event generator in which network managers control the operations of NE manually. Each NESS is designed to be a process of workstation, and the herald and the network event generator are applications of PCs in a network environment underlying the TCP/IP. The simulator has been implemented and successfully used to test a fault management system, which is a kind of TNMS.

I. Introduction

With the rapid advance and increased reliance on telecommunications networks (TNs), network technologies continue to evolve from analog to digital transmission, from the plesiochronous digital network (PDH) to the synchronous digital network (SDH/SONET). Currently, SDH/SONET is one of the most popular and has been chosen as the transmission technology for next-generation protocols, such as B-ISDN[1]. SONET/SDH [1] is a fiber-based synchronous high-speed transmission system that has several advantaged features in the aspects of transmission and operation, administration and maintenance (OAM) capabilities.

SONET/SDH allows several kinds of architectures such as point-to-point, chain/tree, and ring. The one underlying the ring topology is the best to provide the excellent self-healing capability to increase the reliability and the quality of services of network, denoted as *Self-Healing Ring (SHR) network*[2]. It is composed of intelligent network elements (NEs) connected by fibers forming two rings, i.e., the working and protection rings. The NEs can monitor the status of their equipment and the fibers connected to it. When their equipment malfunctions or the fibers are cut off, they can automatically switch to the backup ones to continue their services. This is the automatically protection switching (APS) capability of SHR network for self-healing.

In addition to the APS, an associated network management system must also be built, cooperating with the OAM functions of network to completely ensure the reliability and high-quality of services. The current trend and standard for telecommunication network management system is defined as ITU-T Telecommunications Management Network (TMN) [3, 4, 5], which is based on the ISO network management standards [3, 6]. The principle components of TMN are the TN, the operation systems (OSs), the data communication network (DCN), and the workstation (WS). A TN consists many types of telecommunication and associated supporting equipment each of which is denoted as an NE. The OSs perform the supervising and management functions on the TN using the standard management protocols, i.e., the Common Management Interface Protocol (CMIP) [3], through the public DCN. The network managers at the WS utilize the appropriate user interface to interpret the management information and to manage the network. Therefore, when conducting a TN of SHR, we would like to construct a

*This research is sponsored by the telecommunication laboratories, under contract No. TL-85-3301~3305.

telecommunications network management system (TNMS) conforming to the TMN [8, 9].

To test the functionality of TNMS, a testing environment of SHR network is required to provide the OAM operations and the associated network operations, including the normal and the abnormal. The abnormal operations such as the protection switch on failure or shutdown of network equipment are not appropriate to be performed on a physical network, especially when it is in service. Besides, when the network is under construction, some operations especially those of fault and configuration management are not feasible for testing. Hence, the testing environment is better to base on a simulator of network rather than a physical one. In the simulator, the operations of network are easier to control, observe, thus test. In this paper, therefore, we will design a simulator for the SHR TN to provide the TNMS a development and testing environment. The simulator is used to simulate the OAM and the related functions that support the management functions of TNMS and the testing on them. (Currently, the focus is put on the functions of fault and configuration management, and those of other management areas, i.e., accounting, security, and performance management, are under study.)

For the purpose of supporting the TNMS, the simulator has to provide the network events and states of resources of SHR to the TNMS so that it can proceed the appropriate supervising and management operations. The network events include the alarms such as those due to the abnormal situation of NE and the general events such as the indication of state change of NE. The state of resources includes service, usage, and administration states. The events and states of resources are usually managed by the agents of NE. (In this paper, the NE is referred to the physical telecommunications equipment responsible for the general operations, and the agent is to the software/firmware responsible for the management operations associated with the TNMS.) The agent uses a management information base (MIB)[6] to manage the states of resources, and it exchanges the management information (such as network events) with TNMS using the CMIP. Accordingly, the simulator for the SHR TN must provide the following functions:

1) simulate the functions of agent
2) simulate the MIB at agent
3) provide the CMIP communication interface
4) simulate the event generation by the NE

This paper is organized as follows. Section II presents the background knowledge, especially about SHR network and TMN, and the functional requirement to the simulator. In section III, we will propose an architecture of simulator and discuss the design of its important subsystems. Based on the architecture, we implement the simulator in a Ethernet-based LAN composed of Sun Sparc workstations and IBM compatible PCs. In section IV, an example is shown, that results from a real situation (a failure of network transmission system) in the SHR network using the implementation. Finally, the conclusion is discussed in section V.

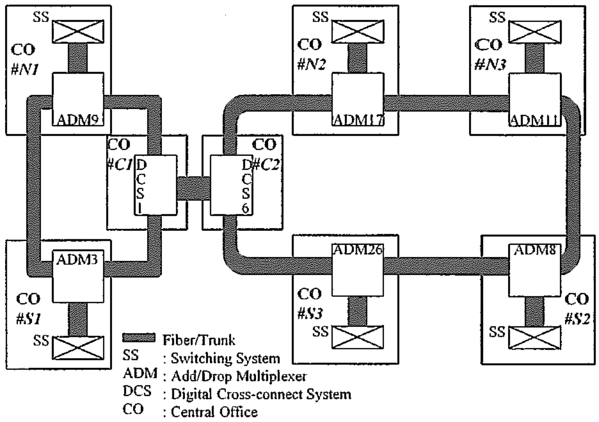

Figure 1 An example SHR network

II. The Background and the Requirement

2.1 The SHR network

Figure 1 is a typical SHR network. For the ease of presentation, the central office (CO) in the figure is simplified to have only a transport network element (NE), i.e., an add-drop multiplexer (ADM) or a digital cross-connect system (CS) [2, 11] and a switch NE, i.e., a switch system. In COs, switch NEs are connected to transport NEs which are connected to each other by the fibers forming SONET rings. Different rings are connected at the COs containing the DCS.

A transport NE is the SONET equipment responsible for the general transmission operations such as transmission, administration, and especially the automatically protection switching (APS). An associated agent is used to manage and maintain its operations and to provide the interface to the TNMS for further management.

SONET is a fiber-based synchronous transmission system defined by the ANSI T1 committee to provide the direct synchronous multiplexing and the standard for signals and OAM (Operation, Administration and Maintenance) operations of networks. The SONET protocols have four layers, i.e., Photonic, Section, Line, and Path, with hierarchical relationship. In the SONET protocol architecture, the basic transmission unit called frame, whose electronically signal format in SONET is STS-N (Synchronous Transport Signal level N) and its corresponding optical signal is OC-N (Optical Carrier level N), includes the SPE (Synchronous Payload Envelop) for the general data and the OH (OverHead), including OHs of section, line, and path, for OAM operations. The overheads in the frame can be seen as the data communication channel (DCC) in which the NEs exchange their OAM information. Every layer, except the photonic layer, has its own set of control signal for the alarm surveillance, performance monitoring and operation control [11].

When the terminal equipment of a certain layer detects a failure, the signals embodied in the corresponding

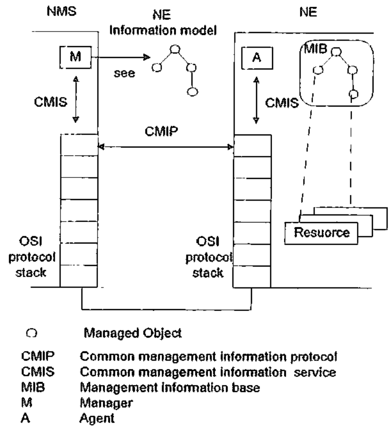

Figure 2 The relationship between agents and mangers in TMN

overhead is transported to the terminal equipment's in up and down streams so the appropriate operation such as the APS can be launched. Besides, the equipment also informs the OS in terms of alarms so that OS can diagnosis the failure and try to recover it.

Besides the signals for communication problems, the vendors of equipment would define the equipment failure signals for the failure of the equipment in the NE. When the hardware in the NE detects the malfunction of its equipment, it will deliver the failure signal to the NE agent who will perform the protection switch and send the alarms to the OS.

In addition to transport NEs, switch NEs would also deliver the OAM signal for the purpose of OAM. However, they are not fully standardized.

So far, we describe the OAM signals and functions supported by NEs. These signals and functions that relate to the network management will be managed and maintained by the agent of NE. The agent is responsible for the manager (the TNMS in our case) to managed the associated NEs. It will accept the management commands from the manager and perform them on NEs, and propagate the notifications such as alarms to the manager. In the following, we will describe the functionality of agents from the TMN point of view, including the functions and information exchange.

The logical relationship between the agent and the manager is shown in Figure 2[5]. The management information at the agent part is logically managed in an object-oriented manner. All the resources to be managed are called managed objects (MOs), including the physical such as network equipment and the logical such as the channel, alarm severity assignment profile. An MO is defined by the following items:

1) attributes visible at the MO boundary
2) system management operations that can be applied to MOs
3) behavior exhibited by MOs in response to management operations

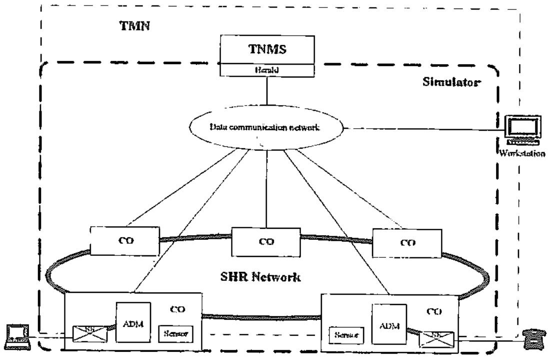

Figure 3 The scope of simulator for SHR network

4) notifications that be emitted by MOs

All MOs are maintained in a conceptual repository, called MIB (Managed Information Base) in a containment tree defining the containment relations of MOs[10]. These MOs are "seen" and managed by the manager via the Q_3 interface [5]. The Q_3 underlies the CMISE (Common Management Information Service Element) including the CMIS (Common Management Information Service) and CMIP (Common Management Information Protocol)[3].

From the functional point of view, according to the ITU-T M.3010, functions of network management can be divided into several blocks, i.e., the Workstation Function (WSF), the Operation System Function (OSF), Mediation Function (MF), Network Element Function (NEF), and the Q Adaptor Function (QAF) [5]. These functional blocks are distributed in the physical components of TMN such as the OS, WS and the NEs. Basically, the WSF, OSF, QAF and NEF are configured in WS, OS, Q adaptor, and NE. The MF can with NE or a specific device called Mediation Device. Therefore, the agents of NE may also support the QAF, NEF and optionally MF.

2.2 Requirement and Assumption of the Simulator

In the last subsection, we briefly describe the functions of NEs and their agents of a SHR network. With them in mind, we define the functional requirement and the assumption for the SHR network simulator as follows:

The SHR network to be simulated is assumed as a single ring network with several COs interconnected with the TNMS via the DCN as shown in Figure 3. (The case of dual rings can be extended by enhancing the functions of NEs and agents without changing the architecture of simulator.) Three types of NEs are in the CO:

1) ***Transport NE.*** This type of NE is a SONET ADM with the transmission rate of OC-3. It can multiplex, demultiplex and relay signals, and provide 28 DS1 channels to the switch NE. It also supports the APS.
2) ***Switch NE.*** This type of NE is a switch system receiving the DS1 channels from transport NE, each of which can support 24 DS0 subscribers.
3) ***Sensors.*** A CO is configured with several sensors to detect the abnormal situation of its environment, such as the fire alarm, or power loss, etc.

The agents of NEs support the NEF and MF to manage the resources of NE as MOs. (QAF is excluded because it is for the adaptation to non-TMN and we considered the TMN only.) The interface between NEs and TNMS is similar to Q_3. (The Q_3 interface is not completely implemented because, in our implementation environment, the communication is achieved of TCP/IP instead of OSI protocol stack.) A herald is with the TNMS for converting the management information between them. Furthermore, for the ease of TNMS operations, the herald also provides several APIs, similar to system management functions defined by ISO[10].

The behavior to be simulated can be divided into the following two categories:

1) the processing of alarms generated by the NE
2) the processing of commands from the TNMS to agents

The alarms to be considered are the following (according to [7]), (1) communication alarms, (2) equipment alarms, (3) environmental alarms, (4) quality of service alarms, and (5) correlative alarms. The processing of these alarms includes their occurrence and clearance. In general, when the NE detects the abnormal situation, it performs the appropriate management operations such as the protection switch or generating the correlative alarms to inform other NEs, and report it to the agent. Then agents store the alarms using the MIB and report it to the TNMS via Q_3 depending on the notification condition associated with the alarm. When the NE detects the clearance of the abnormal situation, it and its agents execute the management operations to mark the situation of alarm cleared. Every type of alarm has its specific processing scenarios for occurrence and clear. (The details of their definitions are referred to [8, 9].)

In addition to the alarm processing, agents have to accept the commands from the TNMS, including (1) query the attributes of MO, (2) set the attributes of MO, and (3) execute the operations of MO. The commands at the TNMS have to be translated into the CMIS requests, and sent to the agent via Q_3. The agent then accepts these requests, performs the necessary management operations to MIB and NE, and returns the response to the TNMS in terms of CMIS responses.

Conclusively, the functions of the simulator are to:

1) accept the commands from the TNMS and interpret the management information from agents to it.
2) simulate the functions of NE and its agent, including the processing of alarms and the management of MIB.
3) provide a user interface to generate the signal from NE. As we don't completely simulate the operation of NE, a user interface is provided to the user of simulator to generate the signals from NE.

III. The Design of Simulator

3.1 The Architecture of Simulator

According to the functional requirements of simulator, we design a system architecture similar to the architecture

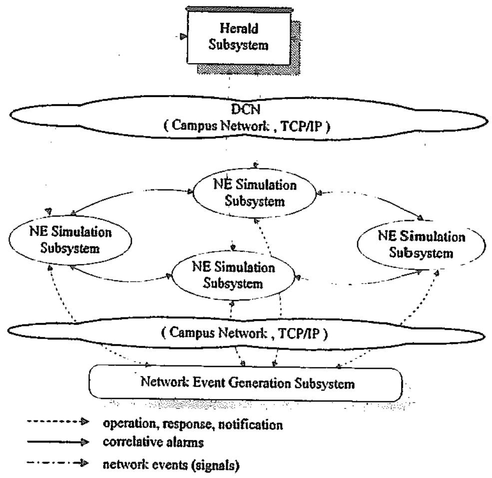

Figure 4 System architecture of the simulator

of physical network as shown Figure 4 with the following subsystems:

1) ***Herald Subsystem (HS)***: It is a herald between the TNMS and the simulator to translating the commands from TNMS into CMIS request and convert the CMIS responses for TNMS.
2) ***NE Simulation Subsystem (NESS)***: It responds to the simulation of three types of NEs, i.e., the transport, switch and sensor NEs, and their agents in a CO.
3) ***Network Event Generation Subsystem (NEGS)***: It provides a uniform user interface for user to setup the signals generated by the NE.

Since the corresponding equipment's of the above subsystems are connected by the DCN in a physical SHR network, each subsystem is implemented by a task distributed in a LAN underlying the TCP/IP and exchanging the following types of information with one another:

1) operations, responses, and notifications between HS and NESS
2) correlative alarms between NESS and NESS
3) network events (signals) between NEGS and NESS

This parallel architecture underlies the facilitated and popular LAN, and has several advantages such as the ease of implementation, the flexibility of extension, and the improvement on performance. In following sections, we will describe and design the functions of each subsystem.

3.2 The Herald Subsystem

The herald subsystem provides an interface to convert the formats of commands/responses between TNMS and the Q_3-like interface. Q_3 is the interface between TNMS and agents of NEs in a physical SHR network, in which the commands/responses is delineated in terms of CMIS requests/responses underlain by the OSI protocol stack, but, in our implementation environment, the underling protocol is TCP/IP. Thus, we can only have a Q_3-like interface instead of Q_3. It provides the requests/responses whose parameters and formats are similar to those of Q_3.

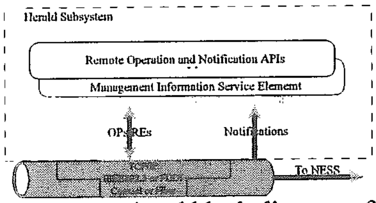

Figure 5 Functional block diagram of HS

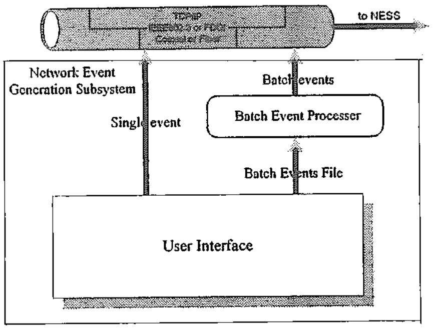

Figure 6 Functional block diagram of NEGS

Besides, as the Q_3 interface has complex formats of parameters, to ease the using of herald, another interface is provided, called *remote operation and notification APIs*. The idea of this interface is similar to the system management functions to collect the frequently-used and important management functions above the Q_3-like interface, and the interface to these functions is simple. Thus, in the herald subsystem, two types of interfaces are provided, the Q_3-like and the remote APIs. The functional block diagram of the herald subsystem is shown in Figure 6 with the following two important blocks:

1) Remote Operation and Notification APIs: 15 APIs so far are provided, i.e., RAPI-SetEFDFilter, RAPI-SetEFDFilterResponse, RAPI_SetASAP, RAPI_SetASAPResponse, RAPI-EFDControl, RAPI_EFDControlResponse, RAPI_QueryAlarm, RAPI_QueryAlarmResponse, RAPI_QueryLog, RAPI_QueryLoogResponse, RAPI_GetAttribute, RAPI_GetAttributeResponse, RAPI_SetAttribute, RAPI_SetAttributeResponse, and RAPI_EventReport [9]. These APIs are converted to CMIS services such as M_Get, M_Set, M_Action, and M_Event_Report, and the CMIS responses are converted to appropriate remote notification APIs.
2) Management Information Service Element: The CMIS services and responses are converted to the TCP/IP socket services, such as to map MO addresses and physical IP address and socket port, and to encapsulate the parameters of CMIS into socket PDU.

3.3 The Network Event Generation Subsystem

The network event in this subsystem is referred to the information about the alarms or signals that are generated by NEs due to the abnormal condition of equipment. As we only simulate the management operations of NE, the simulated NE is impossible to detect the failure of hardware and to generate the events. Thus, network event generation subsystem (NEGS) is provided to the user of simulator to generate the events manually. She can hypothesize a failure of the equipment, and then use the NEGS to generate the events due to the hypothesized failure. The possible situations of event's generation are:

1) generate only a single event
2) generate a group of events whose orders of generation are dependent
3) generate several events that has to occur simultaneously

Thus, NEGS has the functional block diagram as shown in Figure 7, including the following blocks:

1) ***User Interface***: The user interface is for the user to set the events to be generated. She can directly send a single alarm in a dialogue, associated with the alarm type and the options of parameters of that alarm, to set up the parameters; otherwise, She can define a group of alarms and their time relations in a batch event for delivery.
2) ***Batch Event Processor***: This processor is responsible for interpreting the content in the batch event editor and controlling the order and time of event's delivery to NESS.

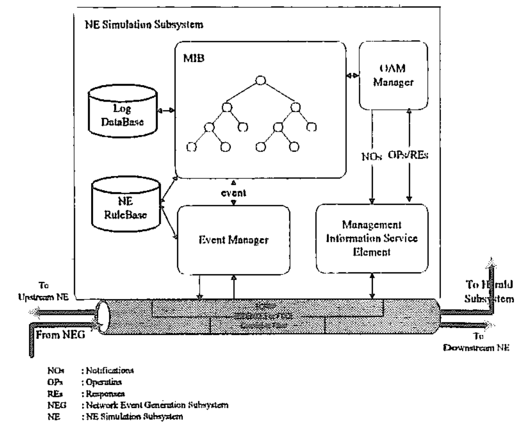

Figure 7 Block diagram of NESS

3.4 NE Simulation Subsystem

This subsystem is the main one of the simulator to perform the management functions of all the NEs in a CO. Functions include:

1) ***Event Processing***: The general scenario for event processing in NE and its agent is described as follows:
   a) The NEGS sends an event to the NESS to denote an abnormal situation in NE using socket services.
   b) According to the event received, NESS performs the associated management operations such as generating correlative alarms on a communication alarm or executing APS on an equipment alarm.
   c) NESS then notifies the MIB to change the values of attribute of MOs relative to this event, and notifies the event to the agent of NE.
   d) The agent receives the event, stores in a log file, and sends it to a filter to verify its notification

condition. If the event satisfies the condition, it is sent to TNMS in the format of M_EVENT_REPORT in CMIS via the TCP/IP socket services.

2) ***Management operations processing***: The general scenario for processing the management operations from the TNMS is as follows:
   a) HS sends a management operation from TNMS in terms of CMIS requests to NESS. NESS receives the requests, and interprets the request to identify the management operation.
   b) On receiving an operation, NESS performs management operations in the relatives MOs.
   c) NESS then returns the result of operations to the TNMS in terms of CMIS responses.

According to the above scenarios, we define the functional block diagram of NESS as Figure 7, including:

1) ***Event Manager***: NESS has to interpret the event from NEGS, do the relative processes and notify this event to the relative MO in MIB.
2) ***NE Rulebase***: The processing of events depends on their type. Thus a rule base is required to maintain the information, such as relative alarms' generation rules and automatically protection switch rules, etc.
3) ***MIB***: MIB is the repository of local MOs. All MOs are maintained in a containment tree so that MIB can (1) query an MO or (2) select one or more MOs on the tree. When an MO is notified by an event, it (1) updates the attribute value or (2) executes the relative management operations. MIB will notify the OAM manager after the processing of events.
4) ***OAM Manager***: It is an intermediate subsystem between the TNMS and MIB. The events from the MOs in MIB have to be recorded, filtered, and sent to TNMS; the operations from TNMS have to be interpreted and notified to the relative MOs
5) ***Log Database***: This is a store to record all events occurred in the CO.
6) ***Management Information Service Element***: This element is associated with the OAM manger to convert the format of events between the one used in NESS and the one in terms of CMIS services. (This one is packaged in terms of TCP/IP socket services on delivery.)

3.5 The MIB

MIB plays a crucial role in the simulation. In this subsection, we describe more detail about MIB, including the containment tree and the functions of MO. According to the characteristics of simulator, the TMN related standards [10], and the scenarios of the operations in the TNs of Taiwan, we design the following classes of MOs corresponding to the physical and logical resources in a SHR network whose containment tree are shown in Figure 8..

Location class: This class of MO is the root of the designed containment tree, representing a CO. Three classes of MOs are contained, i.e., Transport NE, Switch NE and Sensor.

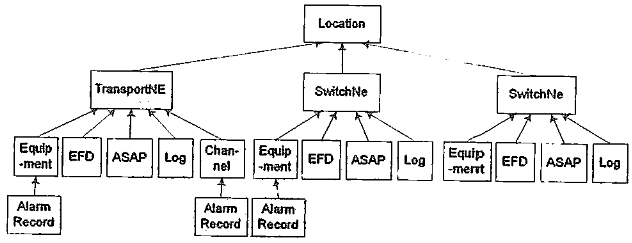

Figure 8 Containment tree in MIB

1) ***TransportNE class***: It corresponds to a physical transport NE in the CO. The MOs contained includes:
   a) *equipment class*: It represents the physical equipment in the transport NE.
   b) *channel class*: It is a channel to transmit the data in the transport NE, such as the DS3, or DS1 channel.
   c) *alarmRecord class*: The MO denotes the alarm detected by the equipment or the signal detector of channel in transport NE. The records can be queried by the TNMS.
   d) *EFD (Event Forward Discriminator) class*: This MO is a logical resource recording the notification conditions associated with alarms.
   e) *ASAP (Alarm Severity Assignment Profile) class*: It records the severity and the cause of alarms in the transport NE and allows network managers to dynamically modify them.
   f) *Log class*: This class denotes a permanent logging of alarms generated and states changed.
2) ***SwitchNE class***: It corresponds to a physical switch system. Classes included are:
   a) *equipment class*: It represents the physical equipment in the switching NE, such as communication modules, switching modules, etc.
   b) *alarmRecord class*, *EFD (Event Forward Discriminator) class*, *ASAP (Alarm Severity Assignment Profile) class*, and *Log class*.

   The other four classes of MOs have the similar meanings and definitions of those in transporNE.
3) ***Sensor class***: This class corresponds to the sensor configured in the environment of CO to detect the abnormal situation of environment. Classes include a) *alarmRecord class*, b) *EFD (Event Forward Discriminator) class*, c) *ASAP (Alarm Severity Assignment Profile) class*, and d) *Log class*. These classes of MOs also have the similar meanings and definitions of those in transporNE.

The above MOs have their common and specific attributes, behavior, notifications and operations[9]. For example, the class name and the instance ID are common attributes, and the AS(administration state) is the specific one of EFD class. Common operations include the query the MO attributes, and the specific one is such as the MO of comAlarm to notify the generation of communication

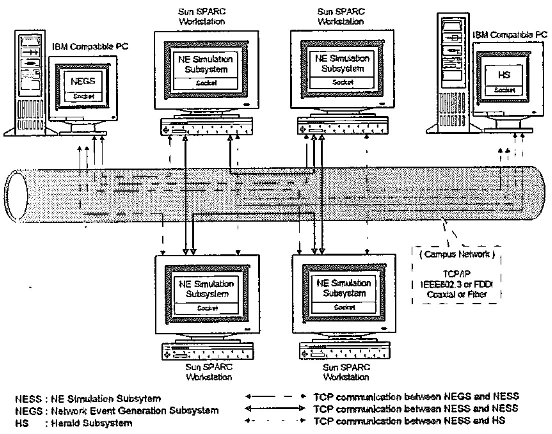

Figure 9 Hardware configuration of the simulator

alarms. In addition to the operations of MOs, the MIB has to provide the following two operations:

1) *the query to a specific MO*. The MIB uses distinguished names, class paths and the object identification tree (OID tree) along with the containment tree to provide a method to identify a specific MO. This is the basic function of MIB [10].
2) *the query to a group of MOs*. This operation is to allow the TNMS to perform the operations to several MOs with a single operation. The MIB uses the parameters of base object class, base object instance, scoping, and filtering to identify a group of MOs in the containment tree.

IV. Implementation and Execution Example

The simulator has been implemented in an Ethernet-based LAN composed of SUN Sparc workstations (whose OS is SUN OS, a UNIX) and IBM compatible PCs (whose is Windows 95). Each NESS is implemented as a UNIX process, and the HS and NEGS are as Windows 95 applications. We use the TCP/IP socket services to implement the communications between subsystems. An example configuration is shown in Figure 9.

For example, consider the following situation: In the SHR network shown Figure 10, a communication alarm, i.e., LOS (loss of signal) of the line layer occurs and is detected by the ADM at #L1 CO. Then the following four communication alarms are generated by ADM:

No.	Equip-ment ID	direction	chan-nel	cause	payload type	CO	Setup/ Cleared
1	OC-3	East	1	LOS	STS	L1	Setup
2	STS-1	East	1	LOP	STS	L1	Setup
3	STS-1	East	2	LOP	STS	L1	Setup
4	STS-1	East	3	LOP	STS	L1	Setup

On receiving them, 29 communication correlative alarms and 13 events of state changing successively occur. To simulate the situation, a simulator like Figure 9 is built by manually invoking four NESS processes at four workstations. (They can also be built at one workstation as four

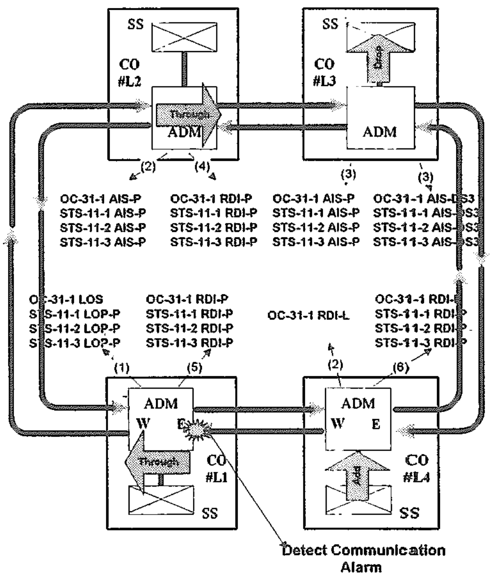

Figure 10 An communication alarm example

processes depending on the physical circumstance such as the availability and the load of workstations. The configuration of other NESSs, NEGS, and HS is set in the parameters to the NESS process.) NEGS and HS applications are at PCs. (The configuration is read in a configuration file.) Then the user writes a batch file containing the first four alarms in the batch editor, and demand the NEGS to sent them to the NESS of L1. On receiving the alarms, the NESS of L1 automatically generates the correlative alarms and events to other NESSs and reports them to the HS, and so do other NESSs. Since all these 46 alarms have to be sent to OS, the simulation result can be seen at HS as shown Figure 11. Furthermore, we have also linked the HS with a fault management system to successfully diagnose the failure at L1 and other hypothesized faults, which is detailed in [13].

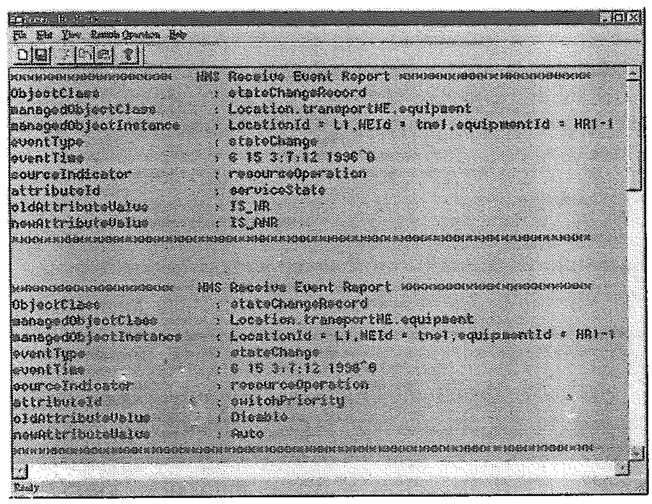

Figure 11 The result (partial) of the example at HS

V. Conclusion

The SHR network is the TN of the new generation with high bandwidth and reliability. In this paper, we design a SHR network simulator with the following functions:

1) Provide the remote notification and operation APIs in the herald for the TNMS
2) Provide a Q_3-like interface for a TCP/IP-based DCN
3) Simulate the agents of NE, and their communications
4) Provide the MIB in the NE
5) Provide a network event generator to allow the user to generate the events of NE manually.

The above functions are provided via three subsystems, i.e., NESS, HS, and NEGS. These subsystems can be freely allocated to the processes of workstations, or applications of PCs in any network environment underlying the TCP/IP. This free and parallel allocation of subsystems gives the user of the simulator great flexibility to configure the simulator according to the SHR network to be simulated and the physical network environment she possesses. Besides, the HS and NEGS allow the user to easily control and observe the operations of the simulator.

We have implemented the simulator on an Ethernet-based LAN composed of Sun Sparc workstations and IBM compatible PCs. It has been used as a testing and underlying environment for a fault management system, which is part of a TNMS, to provide the associated events and alarms for a specific fault to test its diagnosis functions and to provide the necessary operations to test the function of fault recovery [13]. All test cases were successfully executed via the simulator. In addition, we are currently working on using it as the testing environment of other management systems and platform for developing integrated network management systems.

We have designed a basic architecture of the simulator that can simulate most operations of NE and agents, and defined the MOs supporting most functions of fault and configuration management. The functions of simulator can be further extended as follows:

1) Provide the functions of other management areas. A complete TNMS includes the functions of five management areas, i.e., accounting, configuration, fault, performance, and security management. The MOs supporting the other areas should also be incorporated into the MIB to support the operations of their areas. (The security issue is especially critical because the DCN defined in TMN is public.)
2) Provide an interface to generate the new class of MO. Currently, all classes of MOs in MIB must be specified during the design period, and the design and the codes of MIB must be refined to incorporate any new class of MO. This is not an efficient approach. Thus, the simulator has better to provide the capability to dynamically add new classes of MOs.

Reference

[1] Balaji Kumar, Broadband Communications: A Professional's Guide to ATM, Frame Relay, SMDS, SONET, and BISDN, NY: McGraw-Hill, Inc., 1994
[2] Bellcore, SONET Add-Drop Multiplex Equipment Generic Criteria for a Unidirectional, Path Protection Switched, Self-Healing Ring Implementation, Bellcore TA-TSY-000496 Issue 3, August 1990
[3] William Stallings, SNMP,SNMPv2,and CMIP The Practical Guide to Network Management Standards, Massachusetts: Addision-Wesley, 1993
[4] Salah Aidarous and Thomas Plevyak, TN Management into the 21st Century: Techniques, Standards, Technologies, and Applications, NY: IEEE Press and IEE, 1994
[5] ITU-T M.3010, "Maintenance: Telecommunications Management Network-Principles for a Telecommunications Management Network," 1992
[6] ITU-T Rec. X.700, "Information Processing Systems-Open Systems Interconnection-System Management Overview," 1991
[7] Chyan-Goei Chung, et al. Platform and Tools for Developing Integrated Network Management System, Telecommunication Laboratory Project Report (TL-83-3301~3305), 1994
[8] Yeong-Shyan Sun, The Design of the Self-Healing Ringing Simulator for Telecommunication Network, Ms. Thesis of Institute of Computer Science and Information Engineering, National Chiao Tung University, 1995
[9] Ge-Lian Chao, The Design and Implementation of the Simulation system for the Self-Healing Ring TN, Ms. Thesis of Institute of Computer Science and Information Engineering, National Chiao Tung University, 1996
[10] ITU-T Rec. X.720~X.745, "Information Technology-Open Systems Interconnection-Management Information Services-Structure of Management Information: Management Information Model," 1991
[11] Bellcore, Synchronous Optical Network Transport Systems: Common Generic Criteria, BellCore TR-NWT-000253 Issue 2, December 1991
[12] ITU-T M.3020, "Generic Network Information Model," 1992
[13] Chyan-Goei Chung, et al. Platform and Tools for Developing Integrated Network Management System III, Telecommunication Laboratory Project Report (TL-85-3301~3305), 1996