SANDIA REPORT
SAND82—0506 • Unlimited Release • UC—32
Printed April 1982

The Implementation of a Terminal Switching Network Supervisor: SAURON, An Interactive Extension of PACX IV Control Techniques

Michael O. Vahle, Lawrence F. Tolendino

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

SF2900Q(8-81)

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND--82-0506

DE82 015666

SAND82-0506
Unlimited Release
Printed April 1982

THE IMPLEMENTATION OF A TERMINAL SWITCHING NETWORK SUPERVISOR: SAURON, AN INTERACTIVE EXTENSION OF PACX IV CONTROL TECHNIQUES

M. O. Vahle and L. F. Tolendino
Computer Communications Design Division 2648
Sandia National Laboratories
Albuquerque, NM 87185

ABSTRACT

The rapidly growing number of interactive terminals at Sandia National Laboratories which compete for a limited number of computer ports has given rise to the development of a Terminal Switching Network. This paper describes a minicomputer-based supervisory node which interacts with the switches comprising the network and the operators. The supervisor amplifies the control capabilities of the operators, provides a realtime display of the system status, and records usage statistics.

Table of Contents

THE IMPLEMENTATION OF A TERMINAL SWITCHING NETWORK SUPERVISOR: SAURON, AN INTERACTIVE EXTENSION OF PACX IV CONTROL TECHNIQUES

Introduction

Sandia National Laboratories has experienced a rapid growth of interactive computer terminals over the past several years. This growth, coupled with the Laboratory-wide dispersion of interactive computer systems, created two data communications problems. The first problem involves terminal contention for a limited number of computer ports and the second involves terminal switching among various computer resources. In the past, these problems were solved with patch panels and hardware switches. Such solutions were very inefficient and wasteful of precious resources. Circuit switching techniques were chosen to solve these problems.[1]

Using circuit switching techniques, a terminal switching network was designed that provides computer port contention and resource selection. The network consists of multiple switches dispersed over a wide area. Such a distributed network requires a facility to centralize control and management. The central facility allows tailored control procedures reflecting operational needs and provides a convenient location to gather and log system statistics.

This paper describes the central node developed to supervise the terminal switching network. In particular, this paper describes the limitations placed on the supervisor by the switches, the hardware used, the software strategies developed, and the management tools that evolved.

The Terminal Switch

The terminal switching network (TSN) consists of individual Gandalf PACX IV switches. A Gandalf PACX, or port-contender, is an intelligent, programmable, time-division multiplexer. Each switch is restricted to support 512 terminals which can be connected to any of 256 computer ports. These connections can be asynchronous up to 9600 bps or synchronous up to 19.2 kbps. High speed programmable cross point memory controlled by a microprocessor maintains the mapping of the multiplexed data paths between active terminals and computer ports.

Some of the features supported on these switches include:

1. Restricting terminal access to predetermined resources.
2. Reporting all activities over external data paths.
3. Accepting operator commands.
4. Speed transparent interconnects between terminal and ports.

Additionally, the switch provides a standard interface (RS-232, 4-wire, etc.) for the interconnection of dissimilar terminal and computer hardware.

The PACX IV has two RS-232 ports for external control. One port is used to report activities as they occur. The other port is used to submit commands and interact with the microprocessor which controls the switch. For example, this port is used to break a terminal-port connection or to determine the status of a particular port. Using this command interface, the operator can submit commands to set time, to disconnect terminals or ports, to enable/disable ports, to reconfigure ports into groups, and to request system status.

Ports attached to a switch are grouped by service into classes. For instance, ports 10-13 serving computer A are grouped into service class 3. Any terminal requiring computer A can request class 3. If the terminal is cleared for access to class 3 and a port is available, then the connection is made. If the terminal is not cleared, then the connection is denied. On the other hand, if the terminal has access rights to the requested class of service and all the ports assigned to this class are busy, then the terminal can be queued until a port is available.

Implementation

A DEC PDP-11/34 minicomputer system running the RSX11-M operating system was chosen as the TSN supervisory node. The PDP-11/34 system was chosen for its ability to handle hardware interrupts in an efficient and timely manner. This ability is imperative because of the large number of communication lines from the switches interrupting at high rates. In addition, multiplexer hardware is available for the PDP-11/34 to efficiently connect the many communication lines to the node. The RSX11 operating supports multitasking and is ideal for realtime control because of its flexible asynchronous processing ability. There are also many convenient software development tools available and system support routines in RSX11. In short, the milieu provided by the PDP-11 and RSX11 is ideal for this application.

SAURON Software

The functions that the network supervisor performs include:

1. Communicate with the microprocessor that controls each switch.
2. Maintain realtime connection tables and provide current information to system operators.

3. Record port and terminal usage statistics on disk.
4. Maintain a display of system status and problems.

Network operators interact with SAURON through several display consoles. These consoles are connected to the PDP-11/34 over asynchronous communication lines. A summary of network status as well as response to operator inquiries are displayed on these consoles.

The PDP-11/34 is connected to each switch over asynchronous communication lines connected to the switch's console and statistics ports. As the switch connects or disconnects a circuit, the information detailing the action is transmitted over the statistics port. Software called SAURON in the PDP-11/34 intercepts this information, decodes it, and then updates its disk resident statistics tables and memory resident connection tables. The communication line connected to the switch's console port is used to send control messages such as a disconnect or status request between the PDP-11/34 and the microprocessor controlling the switch.

The software, SAURON, is event driven. At startup, a number of asynchronous service routines are initialized waiting to be triggered by input from the switches or operator terminals. When input is received, the asynchronous routine first places the data into a buffer and then queues the buffer for processing. The input routine then signals the control segment of SAURON that an event has occurred, primes itself for additional input and becomes inactive until retriggered. Similarly, the control segment of SAURON is normally in a quiescent state waiting for input to process. To insure the control segment periodically runs, even if the asynchronous routines haven't signaled, a continously running timer that wakes the control segment periodically. The control segment, when activated, completes any activities needing attention and then resets its alarm clock and becomes inactive again. This methodology minimizes the amount of CPU time required by SAURON.

All data processed by the various routines that comprise SAURON is recorded in fixed length buffers. These buffers are drawn from a common pool and returned to the pool when no longer needed. In order to pass information between routines, only the buffer number has to be named. This method leads to very fast and efficient data transfer between the software modules. The mechanism used to manage the buffer space is detailed in SAND80-0821.

Due to the interrupt nature of the asynchronous routines, an interlock mechanism was developed to preclude the possibility of entering FORTRAN subroutines recursively. The scheme uses event flags and RSX11 scheduling of asynchronous routines to alleviate the need for reentrant FORTRAN subroutines.

Since SAURON interacts with the switches through the console and statistics ports, it is necessary that the software emulate the interactions that a human operator would have with the switch. In particular, since the switch prompts for information as needed, SAURON must interpret these prompts in the context of the current interaction. In the case that SAURON determines that the prompts are inconsistent within the state of the current conversation, SAURON terminates the conversation. Once the conversation is terminated, SAURON tries to reinitialize the conversation and retry the aborted communication. This method of communication was chosen to avoid changing the switch microprocessor code.

Operator Interface

There are several video data terminals (VDTs) connected to the PDP 11/34 as operator interface consoles which serve three functions. First, they provide a dynamic display of system status; second, they are used to input commands to the switches; and third, they are used to display the status of a particular switch.

The dynamic display shows the activity level of each switch in the network as the number of active connections and the percentage of switch capacity being used. These percentages are displayed as a bar graph and are updated in realtime. The display also contains the last error detected on each of the active switches. If a detected error is serious and needs immediate attention, the appropriate error message is flashed until cleared by the operator. On some error conditions, SAURON will automatically send commands to the offending switch that disables all ports and disconnects all active terminals. The time of day is also included in the displayed so that the operator can determine if SAURON is functioning.

Using the VDTs, the operator can submit a variety of commands to the switches. The commands are documented in Appendix I. Some commands exist so that an operator can obtain the current status of ports and terminals. These commands generate a response showing current terminal port connections, terminals queued for service, status of ports, and the data rate of active connections. Variations of these commands allow the operator to display a particular terminal or port, a class of service, all active connections, or the sequence of individual switch connections that comprise an active circuit through multiple switches.

The RSX11 system help facility has been used to provide operator documentation for the commands. This help facility is implemented by interfacing the SAURON software to the RSX11 executive through the intertask communication mechanism. Specifically, the operator request is passed to RSX11 as if it originated from an RSX interactive terminal.

Automatic Control Functions

There are several automatic control functions which take place during TSN operations which do not involve operator interaction. The first of these functions is the continuous polling of all switches in the TSN. The poll requests the status of a computer port. The response to the poll serves three purposes. It verifies the control path between SAURON and the switch, serves to update internal SAURON tables after startup, and provides a security check during normal operations. If a discrepancy is detected, action is taken to break the errant connection and Operations personnel are notified. All the ports on each switch are polled in sequence.

Another automatic function is the submission of a series of commands at a specified time of day. This function is primarily used to redefine access classes on a switch in order to handle time-varying traffic loads. These sequences of commands are contained in disk files and are acted on by SAURON as if they are entered from an operator console. In addition, command files can be submitted by operators. This capability is used when long predetermined series of commands are needed such as after a computer host failure.

The last automatic control function of note is the configuring of a switch that has just recovered from a power failure. For security reasons, such a switch will powerup with all ports disabled. Upon notification of the powerup, SAURON submits a predefined configuration command file which assigns computer ports to service classes and enables certain ports.

Network Management

As part of normal operations, SAURON creates data files which show terminal and port activity. These files include data recording classes of service a terminal connects to, the length of connect time, the number of connects, and the number of times a requested resource is busy. Another file is used to record the service a computer port provides in terms of number and duration of connects. Several programs have been developed to analyze the data collected by SAURON. One program generates reports detailing network usage.[3] In addition to the reports detailing historical usage, analysis reports can be generated that predict the response of the network to varying parameters such as traffic loads and number of ports assigned to a class. Another program can validate the predictions by simulation.[5] Together these programs provide tools that not only categorize network performance, but allow management to test new configurations based on actual data.

Conclusions

The supervisory node, SAURON, has proved to be a valuable aid to Operations personnel in the day-to-day running of the Terminal Switching Network. In particular, the personnel can now respond to user requests in a more timely manner by displaying and analyzing the circuit in question. In times of trouble, such as a host computer failure, the operators can accomplish quick and orderly shutdown of the affected service. Additionally, the network can be easily reconfigured to time varying traffic loads.

The management tools allow planning for future expansion and load balancing. By using these tools, developing problems can be detected and corrected before causing any problems. Additionally, the impact of adding new services can be anticipated and potential conflicts eliminated.

While in the main, SAURON has been an effective strategy to supervise multiple switches; it doesn't represent the ultimate solution. SAURON as presently implemented suffers from several inadaquacies. Since SAURON resides in one minicomputer, it can be interrupted by single failures. Because it is heavily relied upon for all TSN operations, this limitation looms large. Due to inherent limitations in a single minicomputer, neither the number of switches nor the number of operator terminals can be increased beyond the current number. Therefore, growth of the network supervised by SAURON is limited. Finally, connecting SAURON to the switches using the switch console port has limitations. This connection has limited functionality since SAURON really can't imitate a human.

APPENDIX I

SAURON COMMANDS AND ERRORS

TIME	ENABLE	DISABLE	CONNECT
DISCONNECT	ALL	PORT	TERMINAL
CLASS	TRACE	PAINT	RESET
LOAD	HALT	ERROR	CLEAR
SEND	SUBMIT	UNPAINT	QUEUE

THE QUEUE COMMAND--QU,n
where n is a legal PACX number

This command reports the terminals queued on the indicated PACX. It's execution is time consuming.

THE UNPAINT COMMAND--UNPA

This command clears the display and prevents the top of the screen from being used to show connection statistics. A PAINT command will put the screen back to normal mode.

THE SUBMIT COMMAND--SUB,filename

This command submits a command file on disk to SAURON. If the command fails to execute a blinking message, 'SUB ERR' will appear. The specifics of the error will be logged on the system console. Standard RSX 11-M naming conventions are used for filename.

THE SEND COMMAND--SEND,n,message
where n (0-3) is a terminal number
message is up to 48 characters

This command allows an operator to send a message to another SAURON operator console. The terminal numbers are assigned as follows:

0 = system console
1 = 880 TCC
3 = 802 TCC

THE CLEAR COMMAND--CLEAR,n
where n (0-5) is the PACX number

This command will clear the warning entry for the specified PACX. Only the NO COMM error is self-clearing except in the case where a second error will overwrite the first.

THE TIME COMMAND--TIME,n
where n (0-5) is the PACX number

This command will set the indicated PACX time to the time maintained in the system supervisor, SAURON.

THE ENABLE COMMAND--ENAB,n,pppp or ENAB,n,pppp-rrrr
where n = PACX number (0-5)
pppp = a port number (0000-0737)
rrrr = a port number (0000-0737)
used for a range of ports

This command is used to enable a PACX port or range of ports.

THE DISABLE COMMAND--DISA,n,pppp or DISA,n,pppp-rrrr

where n = PACX number (0-5)
pppp = port number (0000-0737)
rrrr = port number (0000-0737) used for a range of ports

This command is used to disable specific PACX ports or range of ports.

THE CONNECT COMMAND--CONN,n,pppp,tttt

where n = PACX number (0-5)
pppp = port number (0000-0737)
tttt = terminal number (0000-1773)

This command connects the specified port and terminal on PACX n. Note: The terminal must be requesting service.

THE DISCONNECT COMMAND--DISC,n,Pxxxx or DISC,n,Pxxxx-vvvv
DISC,n,Ttttt or DISC,n,Ttttt-rrrr

where n = PACX number (0-5)
xxxx,vvvv = port numbers (0000-0737)
tttt,rrrr = terminal numbers (0000-1773)

This command is used to disconnect the specified ports or terminals on PACX n. Note that ranges of ports or terminals may be used.

THE ALL COMMAND--ALL,n

where n = PACX number (0-5)

This command will display all the active connections on the indicated PACX.

THE PORT COMMAND--PORT,n,pppp

where n = PACX number (0-5)

pppp = port number (0000-0737)

The PORT command will display the indicated port if any active connection is present.

THE TERMINAL COMMAND--TERM,n,tttt

where n = PACX number (0-5)

tttt = terminal number (0000-1773)

The TERMINAL command will display any active connection on the indicated terminal.

THE CLASS COMMAND--CLASS,n,ccc

where n = PACX number (0-5)

ccc = class number (000-177)

The CLASS command will display all the ports assigned to the indicated class.

THE TRACE COMMAND--TRA,n,Ttttt or TRA,n,Ppppp

where n = PACX number (0,5)

tttt = terminal number (0000-1773)

pppp = port number (0000-1773)

The TRACE command will display an entire cascaded PACX circuit. The port or terminal number indicated in the command may be any part of the circuit.

THE PAINT COMMAND--PA

The PAINT command will redraw the SAURON display.

THE RESET COMMAND--RESET,n,pppp

where n = PACX number (0-5)
pppp = port number (0000-0737)

This command will continue PACX polling at the indicated port, pppp. It is useful in determining the effectiveness of previously issued commands.

THE LOAD COMMAND--LOAD,n,pppp,ccc or LOAD,n,pppp-rrrr,ccc

where n = PACX number (0-5)
pppp,rrrr = port number (0000-0737)
ccc = class number (000-177)

The load command sets the indicated ports to the desired class.

HALT

The HALT command will close all open files and stop SAURON. However, SAURON will restart in one minute or less if the task is not removed from the system.

SYSTEM ERRORS

The TSN will report the following errors:

TIME ERROR--1
CONNECT ERROR--2
DISCONNECT ERROR--3(TERM), 4(PORT)

ENABLE--5

DISABLE--6

QUERY--7 STATUS REQUEST TO PACX FAILED

LOAD--8 SET CLASS FAILED

APPENDIX II

SAURON DATA FLOW

To conceptualize the organization and data flow within SAURON, consider SAURON as composed of input/output routines, report generators, data analyzers, event schedulers, system status tables, and statistics file managers. This appendix describes the relationships extent between these objects. The software listings provide detailed documentation of the individual modules.

The input/output routines for the switches and consoles are implemented as reentrant assembly language subroutines. Each routine can therefore control multiple communication lines asynchronously. There are pure areas within each routine for saving the state of each communication line while the executable code is shared. The bidirectional flow of data between these assembly language routines and the remainder of the SAURON routines is accomplished using shared buffers. These buffers are allocated and restored by special FORTRAN subroutines in SAURON. The scheme is described in Reference 2.

All data input from the switches is analyzed by a FORTRAN routine which checks reasonableness and updates system state tables. Additionally, after some preliminary processing, the data is posted in the appropriate data files. The system state tables reside in a virtual array and consist of entries describing the state of each port in the network. These tables provide the means for passing data between the FORTRAN routines which generate operator reports and update the system status display.

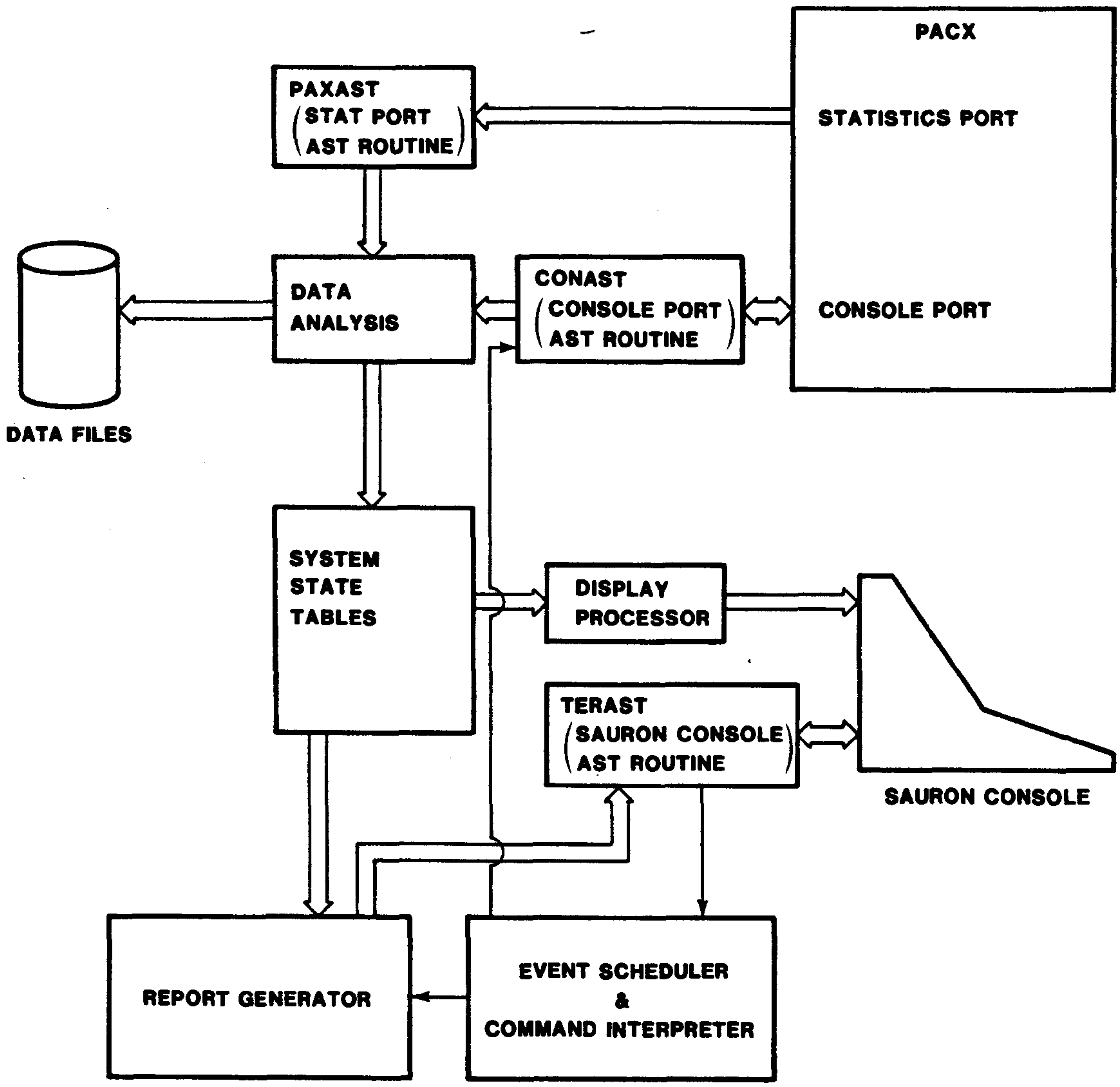
The report generators, which are initiated by operator request, search the system state tables for the requested information, format the data, and pass it to the appropriate output routine. There are a number of these report generators each optimized for a particular type of search. For instance, the routine to output a report showing the status of all ports assigned to a particular class, searches the state tables differently than the routine that traces a connection through multiple switches. The appropriate routine to call is chosen by the event scheduler after the request is decoded.

Many of the FORTRAN routines are organized as overlays to assure that SAURON will meet limitations on task size. However, to provide timely response, all overlays are forced to reside in computer memory instead of on disk. Only the reentrant assembly language input/output routines, the event scheduler, and the buffer management routines are not overlayed. Since the system state tables are large and frequently accessed, they were placed in a virtual array to assure prompt access.

The following figure summarizes the software organization and interaction. The double lines indicate data flow while the single lines represent control paths.

MOV:tmt:6494A:01/29/83X

SAURON SOFTWARE ORGANIZATION



REFERENCES

1. D. J. Gibson, "Long Range Communications Plan for Sandia National Laboratories, Albuquerque, NM," Sandia National Laboratories, Albuquerque, NM, SAND82-0260.

2. M. O. Vahle, and L. F. Tolendino, "An Overview of a Minicomputer Network," Sandia National Laboratories, Albuquerque, NM, SAND80-0821.

3. W. Hauenstein, "Data Analysis of a Terminal Switching Network," Sandia National Laboratories, Albuquerque, NM, SAND81-1749.

4. M. O. Vahle, "The Allocation of Computer Ports within a Terminal Switching Network: An Application of Queueing Theory to Grandalf Port Contenders," Sandia National Laboratories, Albuquerque, NM, SAND81-0176.

5. L. F. Tolendino, "Simulating the Operation of a Port Contender with SIM4: Applying Monte Carlo Techniques to the Grandalf PACX IV," Sandia National Laboratories, Albuquerque, NM, SAND82-0162.

DO NOT MICROFILM

DISTRIBUTION:

2600	L. E.	Hollingsworth
2610	D. C.	Jones
2612	J. A.	Cooper
2612-1	R. A.	Trudo
2614	A. R.	Iacoletti
2620	R. J.	Detry
2630	E. K.	Montoya
2635	P. A.	Lemke
2636	W. F.	Mason
2640	J. L.	Tischhauser
2644	D. M.	Darsey
2645	L. D.	Bertholf
2646	M. R.	Scott
2648	D. H.	Schroeder
2648	M. D.	Thompson
2648	L. F.	Tolendino (5)
2648	M. O.	Vahle
3141	L. J.	Erickson (5)
3151	W. L.	Garner (3)

DOE/TIC (25)
(C. Dalin, 3154-3)