SLAC-PUB-7057
February 1996

Network Resource and Applications Management at SLAC*

C.A. Logg and R.L.A. Cottrell

Stanford Linear Accelerator Center
Stanford University
Stanford, CA 94309

CAL@SLAC.Stanford.Edu
COTTRELL@SLAC.Stanford.Edu

ABSTRACT

The Stanford Linear Accelerator Center (SLAC) has a heterogeneous networked computing environment with a variety of transmission media, protocols, equipment from multiple vendors, Local Areas Network (LAN) and Wide Area Network (WAN) connections, workstations, servers, legacy mainframes, operating systems, network services and applications, and users of various skill levels. New technologies are continually being deployed as they become available. All of these components work together (most of the time) but result in a complex distributed computing environment (henceforth referred to as the ***system***) which requires automated monitoring and management for the maintenance of high quality performance with limited personnel and budgets.

There is no Network Management Station (NMS) product which comes close to doing the job of monitoring and managing the LAN and WAN for SLAC. However, by making use of Ping, Simple Network Management Protocol (SNMP) and its Management Information Bases (MIBs), as well as network applications (traceroute, File Transfer Protocol (FTP), Remote Procedure Calls (RPCs), Remote Shell (rsh), et.al.), an NMS (Netview for AIX), and the accounting and monitoring facilities provided by the server operating systems, the challenge is surmountable.

INTRODUCTION

Subjectively, the ultimate measurers of the ***system*** performance are the users and their perceptions of the performance of their networked applications. The performance of a ***system*** is affected by the physical network plant (routers, bridges, hubs, etc.) as well as by every "com-

Presented at Networld+Interop Engineer's Conference
Las Vegas, Nevada - April 3-5, 1996

* Work supported by Department of Energy
Contract # DE-AC03-76SF00515

puter” and peripheral device that is attached to it, and the software running on the computers and devices. Thus the availability of detailed configuration, fault, and performance information is critical for leveraging the limited personnel available for maintaining the quality of the network.

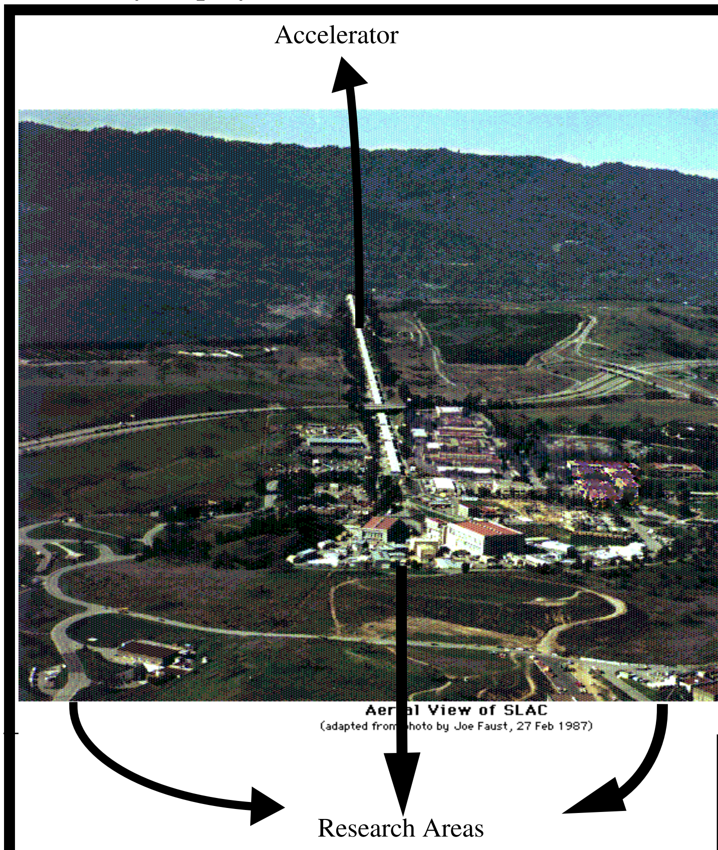
Network Management is generally broken down into five management areas[1]: configuration, fault, performance, security, and accounting management. This paper will discuss how SLAC has made use of ping, SNMP[2], an NMS, and other network services to tackle the configuration, fault, and performance management issues.

OVERVIEW OF THE SLAC NETWORK

SLAC[3] is a national laboratory operated by Stanford University for the US Department of Energy. It is located on 426 acres of Stanford University land, about 40 miles south of San Francisco. SLAC has been in continuous use for over 30 years in a national research program that has made major contributions to the understanding of nature. The Center is one of a handful of laboratories worldwide that stands at the forefront of research into the basic constituents of matter and the forces that act between them. There are also active programs in the development of accelerators and detectors for high energy physics research and of new sources and instrumentation for synchrotron radiation research.

The staff is currently about 1400, of whom 150 are Ph.D. physicists. In addition at any given time, there are typically 900 physicists from other institutions participating in the high energy physics and synchrotron radiation programs.

The SLAC network is physically spread over all the SLAC campus (Figure 1). It is dynamic and continually growing as new research facilities are added at SLAC. Users appear and attach their computers to available network taps, and in fact move their equipment from one network to another without ever notifying anyone. The topology of the SLAC network is changing rapidly to accomodate the growth, and new networking technologies are continually incorporated. Both legacy shared media, ethernet, FDDI, and switched ethernet (10baseT and soon 100baseX) over structured wiring are extensively employed.



The accelerator and research areas have a very unfriendly and noisy environment. There is no air conditioning, it is dusty, there is a lot of RF noise, and there are numerous large power supplies (and magnets) in many areas.

FIGURE 1. Stanford.Linear Accelerator Center

The SLAC network has a large variety of computer hardware, network support hardware, operating systems, and protocols as summarized in Table 1. This makes for a heterogenous and complex network and all the challanges associated with it, including interoperability problems.

.

TABLE 1. SLAC Network Summary

Metric	Value
Number of on-site users	2308 (2644 unique e-mail addr)
Total number of IP Hosts	2655 (3517 IP Addresses)
Macs and PCs on network	1192 (56 Macs on PhoneNet)
Unix Hosts (incl ~75 NeXTs)	427 (63 servers, 161 CAD)
VMS Nodes (VAX+AXP)	202 (25 AXP)
Xterminals	335 (82% NCD[R])
FDDI Rings	15
Routers	12 (60 ifaces)
Switching hubs	4 (2 with RMON)
FDDI Concentrators	11
Bridges	59 (41 w SNMP)
AppleTalk IP Gateways	28
Ethernet probes	74 with RMON
FDDI probes	8 with RMON
Protocol Families	IP (>90%), DEC, Apple, XNS, IPX
Server Operating Systems	Unix (3), VMS, OS/2, WNT, MacOS, VM
Off-site links	BARRnet (10Mbps), ESnet (2@T1, 1@T3)

CONFIGURATION MANAGEMENT

Configuration management basically entails maintaining a knowledge base which contains information on every network support component (cables, routers, bridges, hubs, repeaters, switches, probes, etc.) as well as the computers, printers, and other devices attached to the system. This information includes the manufacturer, model, revision level, medium access control (MAC) addresses, Internet Protocol (IP) addresses, IP names, topology information, contact person, and other protocol addressing information such as for DECNET[R] or Appletalk[R]. In addition it may contain purchasing, warranty, and maintenance information for the various pieces of equipment. This can easily result in the need to maintain a dozen or more pieces of information for every component in the system.

At SLAC an Oracle[R] database called CANDO[4] (Computer and Network Database in Oracle), developed at SLAC, is used to store the above information. However, only devices which are known about can be put in CANDO and that is done manually which is prone to error (people forget, data entry notes get lost or misinterpreted upon entry).

Network support equipment is named based on its location. As problems occur, defective equipment is swapped out and replaced by functionally equivalent equipment, which may not be of the same manufacturer or revision level. The monitoring of the network support equipment (which is described under Fault and Performance Management), often depends upon the manufacturer and revision level. Thus the accuracy of the information in CANDO is critical to the reliability of the automated monitoring and troubleshooting.

Over the past few years as the complexity and size of the network has grown, it has become very difficult to guarantee the completeness and correctness of the data in the database. Thus recently we have begun to use SNMP to fetch the information from the network itself and use it to verify and update CANDO.

For each generic type of device a file is generated daily (from CANDO) which contains the list of the location names, which are also the IP names of the support equipment. Code, specific to each type of device, queries each device (usually once a day) via SNMP for MIB-II information such as the *sysDescr,* number of interfaces (*ifnumber*), *ifPhysAddress, ifDescr, ifAdminStatus, ifOperStatus*, and device type specific information. This is written into tables which are made accessible to the network support people via the World Wide Web (WWW). The tables are also processed and compared to the information on the support devices which is contained in CANDO. Discrepancies are researched (if neccessary) and corrected. The automated verification and correction of the data in CANDO has been very helpful in increasing the reliability of the Fault and Performance Monitoring as well as

[R] indicates a registered trademark

assuring accurate information for trouble shooting.

SNMP is also used to tackle the even larger problem of tracking user application computing equipment which is attached to the network by SLAC visitors without the knowledge of Network Operations, Management, and Development. These user attachments can give sudden rise to various network problems resulting from incorrect attachment and termination, faulty network interfaces, and incorrect network software configuration (e.g. duplicate IP addresses). The network monitoring readily picks up these problems, but they can be hard to track down in a shared media environment if one does not have anything but a MAC address or unregistered IP address and no location information.

To help handle these cases, SNMP is used to read the Address Resolution Protocol (ARP) caches and bridge tables out of the network support equipment such as routers, bridges, and hubs. The information obtained in this fashion is then compared to the data in CANDO. If an entry is not in CANDO, the appropriate local network administrator is asked to provide any information possible. If an item is in CANDO, it is verified and corrected if necessary. In addition, these tables provide information on the approximate location of an offending device.

In addition, the NMS (Netview for AIX[R,5]) provides some configuration change information. Its autodiscovery process creates a log entry which tells when a particular node is attached to the network. This has been useful in correlating sudden network problems to the appearance of an incorrectly attached (or defective) node.

With the availablility of all this information from the network itself, one may ask, "Why do you need a database?" There are several reasons:

1. Not all the information is available from the network.
2. In the heat of the battle, the network may be down, and the information in the various support device tables would be inaccessible.
3. Report generation - Numerous flat files are created from the information in CANDO daily (or more frequently) via cron (UNIX job timing facility) invoked SQL programs. These files are automatically made available for use by the NMS and other network management and monitoring utilities such as a Remote Network Monitoring (RMON) probe Graphical User Interface (GUI), network monitoring data collection, analysis and data reduction programs, which will be described later. Several of the files are also made available by WWW for SLAC general access.
4. To faciliate automated maintenance of name servers and configuration information for Trouble Ticketing applications.
5. To track information on items such as software and non-SNMP'able devices which is not available via the net.
6. The information in CANDO reflects what engineering believes to be the layout of the network and the network support equipment. Printing this information out into various lists for verification (where possible) via SNMP, provides a means for assuring that what engineering believes to be the model, is in fact reality. Various errors in the configuration of some of the more complex network support equipment have been discovered and corrected early as a result of this verification process.

Futures:

CANDO was originally developed to track software purchases and associated information for a few MAC's and some Digital VAX/VMS systems. It worked well for a well organized environment with only a few nodes. However with the advent of inexpensive personal computers and workstations, where everyone purchased and installed whatever hardware and software they wanted, the problem became intractible due to lack of adequate manpower. Hopefully, in the near future it will be possible to find out the hardware and software configurations of every application node attached to the network either via SNMP and a specifically designed MIB, or some other standard mechanism and incorporate this information into CANDO so that when a problem crops up, a complete profile of the ***system*** environment in the area of the problem can be readily obtained.

FAULT MANAGEMENT

System faults most frequently manifest themselves as: slow response, loss of connectivity between nodes, inaccessible file systems, hung processes on computing nodes, and dropped sessions.

Fault management includes the detection of a fault, the diagnosis and correction of the fault, and the tracking of faults and their causes to facilitate proactive maintenance in hopes of preventing their reoccurance.

Some of the common causes of faults and ways to proactively detect and diagnose them include:

Two or more nodes using the same IP address

The most frequent causes of this problem are users passing network software around and failing to change the configuration information in the software, users choosing IP addresses at random rather than obtaining them from an administrator, and the actual incorrect configuration of a node's IP information such as putting the gateway address in for the node's IP address. The connectivity problems resulting from this can be particularly frustrating to track down because of their intermittancy.

Using SNMP to read out router Address Resolution Protocol (ARP) caches which contain IP address to MAC address mappings and comparing them for duplicate IP entries with different MAC address mappings is one way to detect this problem, assuming of course that the offending nodes have some outgoing or incoming communication via the router. Also today, many network support devices and some NMSs have the capability of detecting duplicate IP addresses, and this information is also available via SNMP or often is written to device logs which can be monitored to generate realtime alerts.

Monitoring for duplicate IP addresses has resulted in a significant savings in manpower both for the users and the network support people. By catching the duplicate IP addresses as soon as possible, the local network administrator or users affected can be notified (if the information is in the configuration database), and they can fix the problem before they spend a lot of time trying to diagnose the problem or place a trouble call to the Help Desk which results in a network support person being sent out to investigate the problem.

Physical layer corruptions

The most frequent causes of physical layer corruptions are failing network support equipment such as transceivers, repeaters, bridges, routers, server network interfaces, and user additions and modifications to the network. Other causes include: overheating of equipment during hot spells (not all of SLAC is in a controlled environment), the local fauna snacking on cables, RF interference, accidental loosening of connections, and network interface design errors, just to name a few.

By using SNMP (with the standard and device specific MIBs) data on quantity (utilization and congestion) and quality (number of errors) can be read out on a regular basis from the SNMP'able devices and analyzed. By strategically selecting thresholds on which to generate alerts, many physical layer corruption problems can be detected before the users complain. We have found the effective threshold for errors to be 1 in 10000 packets for our environment. The subject of establishing thresholds is discussed in detail under Performance Management.

RMON has been very useful in diagnosing offending devices and their approximate locations. RMON LAN probes (about 80 of them) are scattered around SLAC with roughly one per shared media ethernet collision domain. Typically when a rise in errors is seen, RMON packet capture is enabled in the relevant probe and the error packets are captured. By knowing the location of the source and destinations in the error packets (from the configuration database and/or bridges tables and ARP caches), it is usually possible to isolate the errors to a given area and equipment set. Typically if the errors only involve one source node, the integrity of that node's interface is checked. If the errors involve packets from several sources then the network in that area is checked for user modifications to the network (physical inspection and testing via Time Domain Reflectometry (TDR)) or for a failing network support device by isolating the suspected section with bridges and adding more LAN probes.

The NMS logs are also checked for recently added devices. Sometimes a rise in errors happens in conjunction with users attaching new nodes to the network. The NMS log which contains messages about newly discovered nodes may show that a certain node was attached about the time the errors began. When all else fails, it is sometimes (but rarely) useful to power off all nodes that can be powered off and turn them on one by one to see when the errors begin.

Intermittent Slowness or Congestion Problems

These problems can be very tricky to diagnose because of their intermittancy. Experience has shown this to usually be the result of network system or applications software which may be running on one or more hosts, and only sometimes related to a physical network layer problem. Here are four examples:

Example 1: Operating System Upgrades

The graphs in Figure 2 show an example of using SNMP to collect data frequently from two network segments with approximately the same range of utilization. On the "problem" segment, the users were complaining about intermittent slowness. There were no errors on the network which would account for it. Normal network monitoring (done once an hour) showed no appreciable spikes in the data. The data displayed on these graphs was collected every 5 minutes over a period of time.

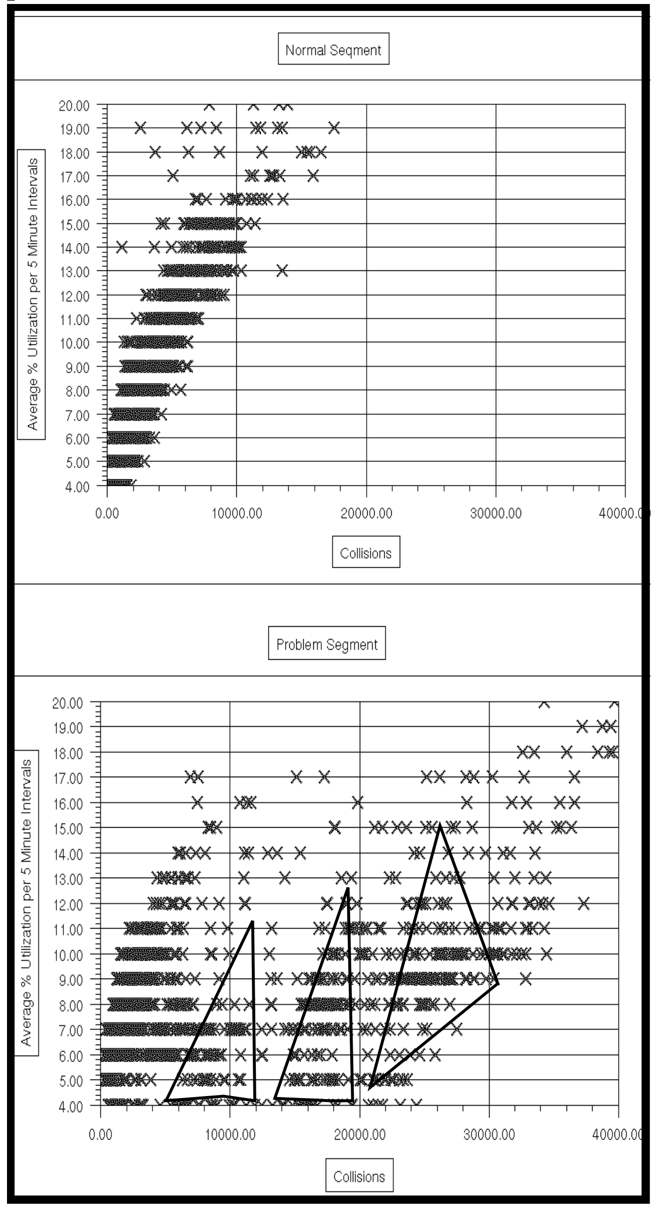


FIGURE 2. Utilization vs Collisions on a Normal Network Segment and a Segment with Problems

Definite structures in the data can be seen in the "problem" segment data (some are "encapsulated" in triangles). Upon closer examination it became apparent the structures were actually, timewise, independent. Via RMON, it was possible to tell which nodes were the top talkers during a couple of the periods. Finally an examination of several of the system logs on the nodes revealed the problem. A new release of an operating system had been recently installed, and the configuration of one of its features which heavily used the network was identified as the cause of the problem.

Example 2: Network Monitoring

At very specific times (20 minutes after the hour) during the day, users (only a few) started complaining of network slowdowns (response time and NFS timeouts). Examination of the hourly network traffic showed no unusual spikes or traffic distribution patterns, although traffic was slightly higher during the day, which was to be expected. The timing of the slowdowns was such that it appeared "obvious" that some time dependent application was a probable culprit. It was decided to investigate the traffic between two specific nodes which seemed to be regulary affected.

An examination of cron tables (UNIX job timing tables) on the two machines turned up nothing. Using SNMP and collecting network traffic data at frequent intervals showed nothing. Finally it was down to examining the performance of the network support equipment on the path between the two nodes. This included router interfaces and bridges. Using a MIB browser to plot and examine some of the performance variables in the vendor supplied MIB every few seconds, it was possible to see that bridges on this path were having numerous well synchronized buffer overflows which corresponded precisely to the network slowdowns. Well, what could be causing that? An examination of the network monitoring readout for fault and performance monitoring failed to show any correlation (it was done at a different time after the hour). However examination of configuration management timing showed that the job which read out the bridge tables was in fact the culprit. Ultimately it was decided that we could no longer afford that luxury of an hourly readout, so the time and frequency were changed and the slowdowns and user complaints disappeared.

Of course the interesting question is why did this problem suddenly crop up when the bridge tables had been being read out for years? Recently the users had made some changes in the location of their workstations and the nature of their work and this resulted in a change in the network traffic patterns generated by these few heavy users. The buffer overflows (which had been going on for some time unobtrusively), finally affected things as they were now in the critical path.

Example 3: Applications Design

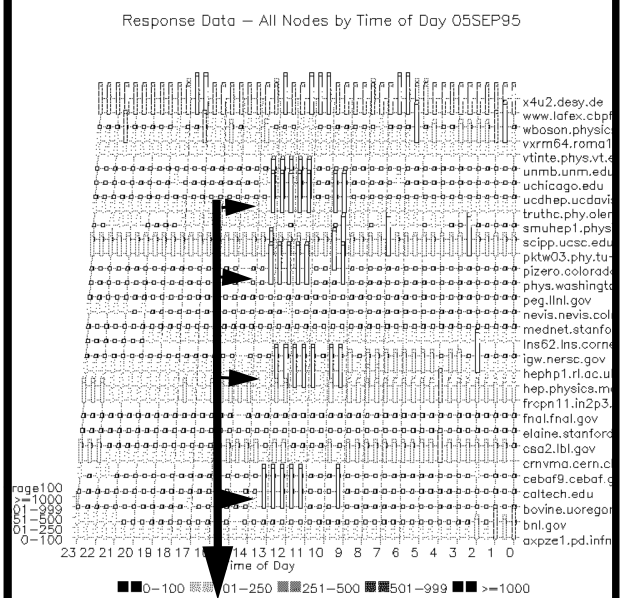
At SLAC a lot of data passes over the network during data analysis. A robust physical network can handle this just fine (no errors and reasonable collision rate), however there may still be complaints of Network File System (NFS) slowness. An investigation into one such case of complaints showed an application's configuration to be at fault. All the data was residing on a file system on the main file server. The analysis applications which ran throughout the day required significant scratch space which was not available on the CPU servers on which they were running. There was plenty of file space on the application's data file system, so large amounts of scratch space were allocated there for the analysis jobs. The actual amount of network traffic due to the reading of the data was small. However investigation of the disk activity on the file server showed the file system for the data to be busy about 75% of the time. The culprit was the analysis program which was thrashing the scratch space (allocated on the file server) while running. The network was handling the data just fine, but the file server was being flogged to death. Subseqent to this, additional disk space was purchased for scratch space on the CPU servers and the application (and others) were redesigned to use local scratch space.

Example 4: WAN Connectivity

There are hundreds of collaborators all over the world who work on experiments at SLAC. Frequently they work from their home research bases with access to SLAC provided by WANs. The natural assumption of off-site users seems to be that if they are experiencing communications problems with SLAC then something must be wrong with the SLAC network. Research has usually shown this not to be true. Ping, traceroute and FTP have been particularly useful in checking out these complaints and providing specific information on slowdowns to the parties responsible for maintaining and providing WAN connectivity as well as absolving the SLAC network of blame.

A list of “critical” off-site nodes is used to ping the nodes of SLAC’s off-site users twice an hour. One ping is done and thrown away to satisify nameservice caching. Next 5 pings of 100 byte packets and 5 pings of 1000 byte packets are done. The values of the packet loss, min, max, and average ping response times are saved in a flat file for analysis.

Figure 3 is a graph of the average response times over the day. Note the period of tall bars which indicate that there was a response time of “>1000milliseconds” for 4 nodes between 11am and 1pm. This was actually the result of a WAN link outage due to a problem a service provider was experiencing. This type of graph is particularly helpful in visualizing correlations in response time problems and outages with user complaints.



4 different nodes lost connectivity due to a service provider outage.
The size of vertical bars represent the response time of the pings. A tall bar indicates a long response time or timeout.

FIGURE 3. Graph of Average Response Time over 24 hours for Critical WAN Links

Table 2 (next page) is the output from a diagnostic routine used to investigate a user’s complaint of slowness. The diagnostic routine which generates this table does a traceroute, pings each node along the path, and generates the data in Table 2. From the table it can be seen that significant packet loss begins at node 192.188.35.1. This information was subsequently relayed to the service provider responsible for that node, and the problem was fixed.

To facilitate debugging by remote users, we have enabled the SLAC WWW server to perform a traceroute to the user’s client node.

.

TABLE 2. Output of Ping and Traceroute Diagnostic Routine

NODE	#pings	#loss	%loss	min	max	avg	#loss	%loss	min	max	avg
192.68.191.2	525	0	0.0%	2	7	2	0	0.0%	5	11	6
134.55.4.226	525	0	0.0%	6	18	9	0	0.0%	21	34	24
134.55.12.98	525	0	0.0%	7	19	9	0	0.0%	23	34	25
134.55.12.65	525	0	0.0%	8	17	10	0	0.0%	26	37	29
134.55.12.225	525	0	0.0%	10	24	13	1	0.2%	34	50	38
192.188.35.1	525	30	5.7%	11	24	15	34	6.5%	37	48	40
128.115.237.254	525	46	8.8%	13	24	17	42	8.0%	43	59	48
128.115.245.2	525	41	7.8%	13	44	21	43	8.2%	45	64	50
GEM1.LLNL.GOV	525	43	8.2%	14	25	18	35	6.7%	52	66	56

Faults for which SNMP and RMON May Not Be Helpful:

Several examples of using SNMP and RMON to track down network faults have been given above, however it is important to realize that they cannot identify all problems. This can be particulary true of network interface problems which result in the violation of the timing specifications for placing data on network media. Most probes and network interfaces can only detect packets which meet the rigid timing and signal level specifications for the network media. If the packets are not within the specification, they may simply be ignored or missed altogether. In these cases when nothing has turned up when examining the network via SNMP (looking at interface statistics, capturing packets, looking at traffic patterns, etc.), the only hope may be a digital scope. Only by placing a digital scope on the network and looking at the signals can a problem like this be detected. And to identlfy the source of the packets it may be necessary to decode the scope signals to pick out the address bits.

Since SNMP statistics often show nothing to check for here, one may have to have an upper level diagnostic program to really identify that the problem the user is complaining about exists. One such diagnostic program may be a bouncing ball which is displayed on a user's machine screen over the network. A second such diagnostic program is a sender program which generates sequentially numbered packets which can be checked for packet loss by a receiver program on another host. If the movement is not smooth on the bouncing ball, or numbered packets are being lost, and there are no statistically detectable network errors or congestion problems picked up by interfaces and probes, then it may be time to haul out the scope.

System Fault Detection and Alarming

Probably the most unobtrusive fault monitoring which can be done is to ping critical servers and network support equipment interfaces on a frequent basis and look for a lack of response. If no response is received from an interface, then retry it a certain number of times. If there continues to be a lack of response, issue an "alarm". Although this can be done independently of an NMS, most NMSs are designed to facilitate this type of alarming. Alarms can be issued in several ways: by telepone paging personnel, sending email messages, poping up message windows on the screens of support personnel, or even sounding audible alarms. At SLAC we do the first three currently.

Pinging is simple, but is not very effective in detecting critical server dysfunctions. Servers can ping just fine, but have operational problems which will only be detected by actually trying to get them to do something. Thus it may be necessary to define tests specific to server functions to really tell if they are operating properly and fulfilling their function. These are described in the Performance Management section.

PERFORMANCE MANAGEMENT

Performance management entails the monitoring of the ***system*** environment for resource utilization, error rates, traffic patterns, congestion, physical layer response time, network services response time, and overall quality of service (availability, MTBF, and MTTR).

SLAC makes use of SNMP, Ping, NMS logs, RPCs, and rsh commands to perform this job.

The general scenario for SLAC performance management[6] entails:

- generating lists of critical nodes, network support equipment, and network services
- performing tests and/or gathering statistics via SNMP, Ping, or other specific tests at regular intervals
- analyzing the data hourly for the current day and nightly for longer terms (fortnightly, monthly, and long term trends) and generating graphs of the data as well as tabular summary reports
- reducing the summary data to Hypertext Markup Language[7](HTML) files which only report on data exceeding certain thresholds and incorporating links to the relevant graphs and raw data for ease of further research.
- reviewing daily the reduced HTML performance reports generated overnight (takes 5 to 15 minutes)
- and, reporting at a 5 to 15 minute daily 9:00am meeting any problems noticed. This meeting is attended by a representative(s) of the systems & server support groups, the network operations and development support groups, interested users, and occasionally upper management. This meeting has proven to be very valuable. It brings together experts of differing disciplines and facilitates communication between them so that all factors relating to the ***system*** can be considered in analyzing problems.

The software supporting this system contains about 16,000 lines of code written in SAS[8], Perl[9], REXX[10], and C. Of this 77% is for the analysis, 15% is for the data collection, and 8% for the data reduction (threshold application) phases. The hard part is not the coding, but understanding the details of the equipment and services and figuring out what exactly needs to be monitored and analyzed.

In all the effort has been about 3 full time equivalent people years.

Altogether SLAC gathers about 4 megabytes of raw data a day in the data collection phase. The analysis and data reduction phases generate about 750 megabytes of graphs and reports. The graphs are currently generated in postscript form, and thus the 750 megabytes could be substantially reduced by generating them in another form.

Data Reduction Thresholds:

The thresholds used in the data reduction have evolved over time and have been set by taking into account the utilization and performance level of the network, as well as the psychology of the network support personnel. For example, when we first started measuring the network performance, traffic was low on the network and many segments had a rate of errors (over 1 in 1000 packets). At that point the CRC and alignment error thresholds were set to 1 error per 1000 packets. After some time, the network was cleaned up and traffic began to increase so the threshold on the errors was adjusted to 1 in

10000 packets. If the thresholds had been set to 1 in 10000 packets initially our network support personnel might have given up in despair. Today the thresholds are:

Ethernet probes:

- crc errors and alignment errors > 1 in 10000 packets
- total utilization on a network over 10% for the day
- broadcast rate > 150 per second
- (shorts+collisions)/good_packets > .10

Bridge interface statistics:

- crc errors and alignment errors > 1 in 10000 packets
- buffer overflows and controller overflows > 1

Router interface statistics:

- total interface input errors over 1 in 10000 packets
- collision rates over 1000 in 10000 packets
- crc errors and alignment errors > 1 in 10000 packets
- buffer overflows and controller overflows > 1
- in q and out q drops and discards > 1
- ignored packets > 1

The thresholds for switches and hubs are currently under development, but will probably be similar. However there are other types of thresholds to apply to switches and hubs such as:

- the total amount of traffic going through the device
- how well the traffic is balanced between the ports

Thresholds for LAN pings are:

- packet losses from ping tests greater than 1% in a day
- average response time greater than 10 milliseconds

Thresholds for WAN ping response time and packet loss are set dynamically daily by calculating a 10 week (excluding weekends) running average and standard deviation. Only values exceeding the 10 week average + 3 standard deviations are alerted. WAN links are beyond SLAC's control. We can only monitor them, report on degradations, and try to convince the WAN link providers that the degradation we see needs to be investigated (and hopefully remedied). The information is also made available worldwide via WWW for the users to see, and perhaps, apply pressure from their end.

SNMP Utilization

Almost all modern network support equipment (routers, bridges, hubs, repeaters, switches, etc.) contain SNMP agents which provide statistical information on the traffic passing through them. In addition, there are sophisticated ethernet and FDDI (and other media) probes on the market.

At SLAC, we read network traffic data[6] from all of these devices hourly 24 hours a day 7 days a week via cron controlled jobs. The data is written into flat files which is processed hourly (for the present day) and nightly for the past two weeks, month, and longer term trends. Graphical displays of the data (present and past) are created for much of the data, and tabular reports summarizing the the data are produced.

Ping Utilization

Ping is used to check the performance of the physical network layer. Lists of critical servers, critical offsite nodes (as described above under Fault Management), network probes, and network support equipment are fetched via SQL from the CANDO database daily. Twice an hour all those addresses are pinged once to satisfy nameservice caching, and then 5 times with a 100 byte packet and 5 times with a 1000 byte packet. The results of these pings (percent of packet loss, minimum, maximum, and average response times) are written to flat files. This data is analyzed hourly and nightly (like the SNMP data) and HTML reports and graphs are generated for perusal.

Figure 4 shows the frequency distribution of the response times of some of the national nodes. It gives a clear characterization of the response times of the different nodes.

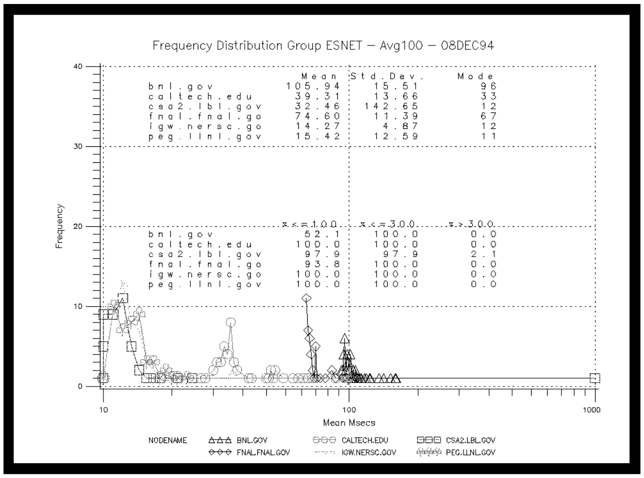


FIGURE 4. Frequency Distribution of Response Times

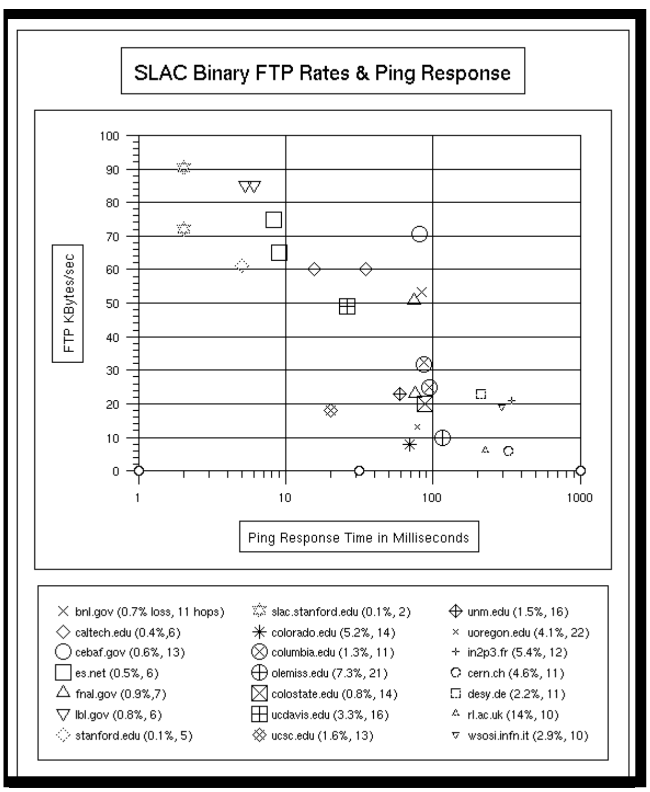


FIGURE 5. SLAC Binary FTP Rates vs Ping Response Times

Figure 5 demonstrates the effect of slow ping response times on FTP rates. The numbers in parenthesis after the node names in the legend are the percent of ping loss and the number of hops.

RPCs and rsh for Monitoring Network Services

Monitoring the performance of the network phyical layer is not really enough The network physical layer can be working fine, but if the servers providing the network services such as email, nameservice, distributed file services, NIS, WWW access, X-terminal and printer services (just to name a few) are not performing to specification the users will undoubtedly claim a network problem exists.

Monitoring network services requires an understanding of the services and the generation of specific tests geared toward each service. In developing the tests it is often necessary to run a few different ones in parallel to see which test will actually detect a problem.

It is important to note that the service tests are not exhaustive or diagnostic tests. They are simple tests which might indicate that a proformance problem may exist with a service.

The current services monitored include:

- email - a simple email message is sent through each mailserver periodically and the length of time it takes to arrive at its destination is recorded and analyzed
- font service response - a request for font service is sent to the various font servers, and the response time is recorded and analyzed
- NFS repsonse - a call is made via /usr/etc/rpcinfo to the various NFS servers and its response time is recorded and analyzed
- SMTP - a routine called xchkaccess[11] is used to test and time SMTP access

- WWW - the routine called xchkaccess is used to test and time WWW access by fetching a specific WWW page
- nameservice - an nslookup request is issued to each nameserver and the response time is recorded and analyzed.

Setting thresholds for network service performance alerting can be tricky. It is frequently necessary to have an individual threshold for each server performing a service due to differences in server memory availability, cpu speeds, and loading configuration. Currently the thresholds are set manually. We are considering whether to automate this by using rolling averages similar to the way the WAN ping thresholds are calculated.

The network services' thresholds are based on the expectations for 50% and 95% of the responses. See Table 3 for an example of SMTP performance reporting. This is the full report for SMTP. Just like other summary reports the network services summary reports are data reduced into an HTML file where only servers/services exceeding the 50 and 95 percentiles thresholds are displayed together with pointers to the raw data file which shows the times the responses were slow.

TABLE 3. :Response Time Summary for SMTP Probe to Port 25 on Dec 14, '95

Node	#samples	ave	50%tile	threshold	95%tile	threshold	#time outs	maxtime	maxsecs
MAILBOX	96	0.497s	0.458s	(<1.000s)	0.657s	(<3.000s)	0	04:20	1.269s
SERV02	96	0.848s	0.616s	(<6.000s)	1.856s	(<15.000s)	0	03:05	10.193s
SERV03	96	0.436s	0.417s	(<6.000s)	0.544s	(<10.000s)	0	08:20	1.497s
SERV04	96	0.253s	0.181s	(<1.000s)	0.706s	(<3.000s)	0	09:05	1.744s
SERV05	96	0.102s	0.029s	(<1.000s)	0.347s	(<3.000s)	0	02:50	1.167s
SLACVM	96	0.117s	0.104s	(<0.500s)	0.228s	(<1.000s)	0	01:36	0.312s

NMS Logs

The logs from Netview for AIX^R are processed daily for information on uptime of the network support equipment and critical server interfaces. The NMS polls at a specified interval and creates log entries when it notices a change of state (up to down, down to up) of a managed server or network support device. This information is used to determine availability, MTBF, and MTTR statistics.

Performance and Utilization Monitoring of *System* Servers

The monitoring of ***system*** servers[12] for performance, utilization, and availablity is also a critical component of our ***system*** monitoring. This is not done via SNMP although. The vendor supplied accounting packages which come with the operating systems are used for this purpose. Again, various reports are generated daily showing utilization and long terms trends which provide the ***system*** server support group the information they need to monitor ***system*** server performance and utilization.

SUMMARY

This paper has discussed how SLAC uses SNMP, Ping, an NMS, traceroute, RPCs, rsh's, and the accounting resources provided by the server operating systems, and even the network services themselves to monitor and manage the performance of various components of the SLAC ***system***. This has all been developed incrementally over a period of 3 years in response to problems that have arisen in maintaining the ***system*** and the need to handle the rapid growth of the ***system***.

One question that frequently comes up is: is all this necessary? We have certainly found it to be very valuable in leveraging our limited manpower and budgets in the maintenance of the SLAC ***system***. It enables us to proactively take care of developing problems before they become disruptive to the user community. The long term trends have been critical in planning for growth and implementing needed expansions before the user community is impacted.

REFERENCES

1. Leinward, Allan and K. Fong, Network Management, A Practical Perspective, Addison-Wesley Publishing Company, 1993.
2. Rose, Marshall, The Simple Book, An Introduction to Management of TCP/IP-based Networks, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
3. The SLAC Institutional Page is available at URL http://www.slac.stanford.edu/
4. Downey, Teresa, "Quick Guide to CANDO Listings" and documentation is available at URL http://www.slac.stanford.edu/usr/local/scs/net/cando/cando.html.
5. IBM, URL= http://www.raleigh.ibm.com/nva/nvaover
6. Logg, C.A. and Cottrell, R.L.A., SLAC-PUB-95-6744, "Network Management and Performance Monitoring at SLAC", Presented at NetWorld+Interop Engineers' Conference, Las Vegas, Nevada, March 1995. Available via FTP from: ftp.slac.stanford.edu, users/cal/interop-paper-3-95.ps
7. December, John and Ginsburg, Mark, "HTML & CGI Unleashed", Sams.net Publishing, 1995.
8. SAS Institute Inc.; *SAS Language Reference,* Version 6, First Edition,Cary, NC.
9. Wall, Larry & Schwartz, Randal; Programming perl, O'Reilly & Associates, Sebastopol, CA, 1990.
10. Uni-REXX, The Workstatation Group,Rosemont, IL, 1994
11. Written by John Halperin at SLAC (JXH@Slac.Stanford.Edu.
12. Charles Boeheim (BOEHEIM@Slac.Stanford.Edu) and http://www.slac.stanford.edu/~boeheim/, and, Lois White (LMWHITE@Slac.Stanford.Edu).

ACKNOWLEDGEMENTS

The authors would would like to thank their many colleagues at SLAC who have contributed to the evolution of this system:

- Lois White (LMWHITE@SLAC.Stanford.Edu) has been responsible for all the SAS programming, and the individual computer system cpu and disk space accounting,
- John Halperin (JXH@SLAC.STANFORD.Edu) has helped with the development of tools to monitor the performance of network services,
- Teresa Downey (TERESA@SLAC.Stanford.Edu) developed and maintains the CANDO database,
- Don Pelton (DEP@SLAC.Stanford.Edu) and David Millsom (MILLSOM@SLAC.Stanford.Edu) of Network Engineering whose physical layer diagnostic experiences are mentioned herein,

and, the other members of the Network Engineering and Network Operations groups.