Network Working Group
Request for Comments: 1147
FYI: 2

R. Stine, Editor
SPARTA, Inc.

April 1990

FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices

Status of this Memo

The goal of this FYI memo is to provide practical information to site administrators and network managers. This memo provides information for the Internet community. It does not specify any standard. It is not a statement of IAB policy or recommendations. Comments, critiques, and new or updated tool descriptions are welcome, and should be sent to Robert Stine, at stine@sparta.com, or to the NOCTools working group, at noctools@merit.edu.

Distribution of this memo is unlimited.

1. Introduction

This catalog contains descriptions of several tools available to assist network managers in debugging and maintaining TCP/IP internets and interconnected communications resources. Entries in the catalog tell what a tool does, how it works, and how it can be obtained.

The NOCTools Working Group of the Internet Engineering Task Force (IETF) compiled this catalog in 1989. Future editions will be produced as IETF members become aware of tools that should be included, and of deficiencies or inaccuracies. Developing an edition oriented to the OSI protocol suite is also contemplated.

The tools described in this catalog are in no way endorsed by the IETF. For the most part, we have neither evaluated the tools in this catalog, nor validated their descriptions. Most of the descriptions of commercial tools have been provided by vendors. Caveat Emptor.

1.1 Purpose

The practice of re-inventing the wheel seems endemic to the field of data communications. The primary goal of this document is to fight that tendency in a small but useful way. By listing the capabilities of some of the available network management tools, we hope to pool and share knowledge and experience. Another goal of this catalog is to show those new in the field what can be done to manage internet sites. A network management tutorial at the end of the document is of further assistance in this area. Finally, by omission, this catalog points out the network management tools that are needed, but do not yet exist.

There are other sources of information on available network management tools. Both the DDN Protocol Implementation and Vendors Guide and the DATAPRO series on data communications and LANs are particularly comprehensive and informative. The DDN Protocol Implementation and Vendors Guide addresses a wide range of internet management topics, including evaluations of protocol implementations and network analyzers.* The DATAPRO volumes, though expensive (check your local university or technical libraries!), are good surveys of available commercial products for network management. DATAPRO also includes tutorials, market analyses, product evaluations, and predictions on technology trends.

* Instructions for obtaining the DDN Protocol Guide are given in Section 7 of the appendix.

1.2 Scope

The tools described in this document are used for managing the network resources, LANs, and devices that are commonly interconnected by TCP/IP internets. This document is not, however, a ''how to'' manual on network management. While it includes a tutorial, the coverage is much too brief and general to serve as a sole source: a great deal of further study is required of aspiring network managers. Neither is this catalog is an operations manual for particular tools. Each individual tool entry is brief, and emphasizes the uses to which a tool can be put. A tool's documentation, which in some cases runs to hundreds of pages, should be consulted for assistance in its installation and operation.

1.3 Overview

Section 1 describes the purpose, scope, and organization of this catalog.

Section 2 lists and explains the standard keywords used in the tool descriptions. The keywords can be used as a subject index into the catalog.

Section 3, the main body of the catalog, contains the entries describing network management tools. The tool entries in Section 3 are presented in alphabetical order, by tool name. The tool descriptions all follow a standard format, described in the introduction to Section 3.

Following the catalog, there is an appendix that contains a tutorial on the goals and practice of network management.

1.4 Acknowledgements

The compilation and editing of this catalog was sponsored by the Defense Communications Engineering Center (DCEC), contract DCA100-89-C-0001. The effort grew out of an initial task to survey current internet management tools. The catalog is largely, however, the result of volunteer labor on the part of the NOCTools Working Group, the User Services Working Group, and many others. Without these volunteer contributions, the catalog would not exist. The support from the Internet community for this endeavor has been extremely gratifying.

Several individuals made especially notable contributions. Mike Patton, Paul Holbrook, Mark Fedor and Gary Malkin were particularly helpful in composition and editorial review, while Dave Crocker provided essential guidance and encouragement. Bob Enger was active from the first with the gut work of chairing the Working Group and building the catalog. Phill Gross helped to christen the NOCTools Working Group, to define its scope and goals, and to establish its role in the IETF. Mike Little contributed the formative idea of enhancing and publicizing the management tool survey through IETF participation.

Responsibility for any deficiencies and errors remains, of course, with the editor.

2. Keywords

This catalog uses ‘‘keywords’’ for terse characterizations of the tools. Keywords are abbreviated attributes of a tool or its use. To allow cross-comparison of tools, uniform keyword definitions have been developed, and are given below. Following the definitions, there is an index of catalog entries by keyword.

2.1 Keyword Definitions

The keywords are always listed in a prefined order, sorted first by the general category into which they fall, and then alphabetically. The categories that have been defined for management tool keywords are:

- the general management area to which a tool relates or a tool’s functional role;
- the network resources or components that are managed;
- the mechanisms or methods a tool uses to perform its functions;
- the operating system and hardware environment of a tool; and
- the characteristics of a tool as a hardware product or software release.

The keywords used to describe the general management area or functional role of a tool are:

Alarm
: a reporting/logging tool that can trigger on specific events within a network.

Analyzer
: a traffic monitor that reconstructs and interprets protocol messages that span several packets.

Benchmark
: a tool used to evaluate the performance of network components.

Control
: a tool that can change the state or status of a remote network resource.

Debugger
: a tool that by generating arbitrary packets and monitoring traffic, can drive a remote network component to various states and record its responses.

Generator
: a traffic generation tool.

Manager
: a distributed network management system or system component.

Map
: a tool that can discover and report a system’s topology or configuration.

Reference
: a tool for documenting MIB structure or system configuration.

Routing
: a packet route discovery tool.

Security
: a tool for analyzing or reducing threats to security.

Status
: a tool that remotely tracks the status of network components.

Traffic
: a tool that monitors packet flow.

The keywords used to identify the network resources or components that a tool manages are:

Bridge
: a tool for controlling or monitoring LAN bridges.

CHAOS
: a tool for controlling or monitoring implementations of the CHAOS protocol suite or network components that use it.

DECnet
: a tool for controlling or monitoring implementations of the DECnet protocol suite or network components that use it.

DNS
: a Domain Name System debugging tool.

Ethernet
: a tool for controlling or monitoring network components on ethernet LANs.

FDDI
: a tool for controlling or monitoring network components on FDDI LANs or WANs.

IP
: a tool for controlling or monitoring implementations of the TCP/IP protocol suite or network components that use it.

OSI
: a tool for controlling or monitoring implementations of the OSI protocol suite or network components that use it.

NFS
: a Network File System debugging tool.

Ring
: a tool for controlling or monitoring network components on Token Ring LANs.

SMTP
: an SMTP debugging tool.

Star
: a tool for controlling or monitoring network components on StarLANs.

The keywords used to describe a tool's mechanism are:

Curses
: a tool that uses the ''curses'' tty interface package.

Eavesdrop
: a tool that silently monitors communications media (e.g., by putting an ethernet interface into ''promiscuous'' mode).

NMS
: the tool is a component of or queries a Network Management System.

Ping
: a tool that sends packet probes such as ICMP echo messages; to help distinguish tools, we do not consider NMS queries or protocol spoofing (see below) as probes.

Proprietary
: a distributed tool that uses proprietary communications techniques to link its components.

SNMP
: a network management system or component based on SNMP, the Simple Network Management Protocol.

Spoof
: a tool that tests operation of remote protocol modules by peer-level message exchange.

X
: a tool that uses X-Windows.

The keywords used to describe a tool's operating environment are:

DOS
: a tool that runs under MS-DOS.

HP
: a tool that runs on Hewlett-Packard systems.

Macintosh
: a tool that runs on Macintosh personal computers.

Standalone
: an integrated hardware/software tool that requires only a network interface for operation.

UNIX
: a tool that runs under 4.xBSD UNIX or related OS.

VMS
: a tool that runs under DEC's VMS operating system.

The keywords used to describe a tool's characteristics as a hardware or software acquisition are:

Free
: a tool is available at no charge, though other restrictions may apply (tools that are part of an OS distribution but not otherwise available are not listed as ‘‘free’’).

Library
: a tool packaged with either an Application Programming Interface (API) or object-level subroutines that may be loaded with programs.

Sourcelib
: a collection of source code (subroutines) upon which developers may construct other tools.

2.2 Tools Indexed by Keywords

Following is an index of catalog entries sorted by keyword. This index can be used to locate the tools with a particular attribute: tools are listed under each keyword that characterizes them. The keywords and the subordinate lists of tools under them are in alphabetical order.

In the interest of brevity, some liberties have been taken with tool names. Capitalization of the names is as specified by the tool developers or distributers. Note that parenthetical roman numerals following a tool's name are not actually part of the name. The use of roman numerals to differentiate tools with the same name is explained in the introduction of Section 3.

free
- arp
- CMIP Library
- CMU SNMP
- DiG
- ENTM
- etherhostprobe
- hammer
- hopcheck
- HyperMIB
- Internet Rover
- map
- netmon (I)
- NETMON (II)
- netstat
- netwatch
- net_monitor
- nfswatch
- nhfsstone
- NNStat
- NPRV
- nslookup
- osilog
- osimic
- osimon
- OSITRACE
- ping
- query
- sma
- SNMP Kit
- tcpdump
- tcplogger
- traceroute
- TRPT
- TTCP

generator
- hammer
- nhfsstone
- ping
- Sniffer
- SpiderMonitor
- spray
- TTCP
- Unisys NCC

HP
- xup

IP
- arp
- CMU SNMP
- Dual Manager
- ENTM
- etherfind
- etherhostprobe
- EtherView
- getone
- hammer
- hopcheck
- Internet Rover
- LANWatch
- map
- Netlabs CMOT Agent
- Netlabs SNMP Agent
- netmon (I)
- NETMON (II)
- NETMON (III)
- netstat
- netwatch
- net_monitor
- nfswatch
- NMC
- NNStat
- NPRV
- OverVIEW
- ping
- proxyd
- query
- SERAG
- Sniffer
- SNMP Kit
- Snmp Libraries
- snmpask
- snmpd (I)
- snmpd (II)
- snmplookup
- snmpperfmon
- snmppoll
- snmpquery
- snmproute
- snmpset
- snmpsrc
- snmpstat
- snmptrapd
- snmpwatch
- snmpxbar
- snmpxconn
- snmpxmon

- snmpxperf
- snmpxperfmon
- snmpxrtmetric
- SpiderMonitor
- SPIMS
- spray
- Tcpdump
- Tcplogger
- Traceroute
- TRPT
- TTCP
- Unisys NCC
- WIN/MGT Station
- xnetmon (I)
- XNETMON (II)
- xnetperfmon

library

- CMIP Library
- Dual Manager
- LANWatch
- proxyd
- WIN/MGT Station

Macintosh

- HyperMIB

manager

- CMIP Library
- CMU SNMP
- ConnectVIEW
- decaddrs
- Dual Manager
- getone
- LanProbe
- map
- Netlabs CMOT Agent
- Netlabs SNMP Agent
- NETMON (III)
- NMC
- NNStat
- osilog
- osimic
- osimon
- OverVIEW
- sma
- SNMP Kit
- Snmp Libraries
- snmpask
- snmpd (I)
- snmpd (II)
- snmplookup
- snmpperfmon
- snmppoll
- snmpquery
- snmproute
- snmpsrc
- snmpset
- snmpstat
- snmptrapd
- snmpwatch
- snmpxbar
- snmpxconn
- snmpxmon
- snmpxperf
- snmpxperfmon
- snmpxrtmetric
- TokenVIEW
- Unisys NCC
- WIN/MGT Station
- xnetmon (I)
- XNETMON (II)
- xnetperfmon

map
- decaddrs
- etherhostprobe
- EtherMeter
- LanProbe
- map
- NETMON (III)
- Network Integrator
- NPRV
- Snmp Libraries
- snmpxconn
- snmpxmon
- Unisys NCC
- xnetmon (I)
- XNETMON (II)

NFS
- etherfind
- EtherView
- nfswatch
- nhfsstone
- Sniffer
- tcpdump

NMS
- CMU SNMP
- ConnectVIEW
- decaddrs
- Dual Manager
- EtherMeter
- getone
- LanProbe
- map
- Netlabs CMOT Agent
- Netlabs SNMP Agent
- NETMON (III)
- NMC
- NNStat
- OverVIEW
- proxyd
- SERAG
- SNMP Kit
- Snmp Libraries
- snmpask
- snmpd (I)
- snmpd (II)
- snmplookup
- snmpperfmon
- snmppoll
- snmpquery
- snmproute
- snmpset
- snmpsrc
- snmpstat
- snmptrapd
- snmpwatch
- snmpxbar
- snmpxconn
- snmpxmon
- snmpxperf
- snmpxperfmon
- snmpxrtmetric
- TokenVIEW
- Unisys NCC
- WIN/MGT Station
- xnetmon (I)
- XNETMON (II)
- xnetperfmon

OSI
- CMIP Library
- Dual Manager
- LANWatch
- Netlabs CMOT Agent
- NETMON (III)
- osilog
- osimic
- osimon
- OSITRACE
- sma
- Sniffer
- Snmp Libraries
- SpiderMonitor
- SPIMS
- XNETMON (II)
- xnetperfmon

ping
- etherhostprobe
- hopcheck
- Internet Rover
- map
- netmon (I)
- net_monitor
- NPRV
- ping
- spray
- traceroute
- TTCP
- Unisys NCC
- xup

proprietary
- ConnectVIEW
- EtherMeter
- LanProbe
- SERAG
- TokenVIEW

reference
- HyperMIB
- Unisys NCC

ring
- ConnectVIEW
- LANWatch
- map
- NETMON (III)
- netwatch
- proxyd
- Sniffer
- Snmp Libraries
- snmpd (II)
- TokenVIEW
- XNETMON (II)
- xnetperfmon

routing
- arp
- ConnectVIEW
- decaddrs
- etherhostprobe
- getone
- hopcheck
- NETMON (III)
- netstat
- net_monitor
- NMC
- NPRV
- query
- Snmp Libraries
- snmproute
- snmpsrc
- snmpxrtmetric
- traceroute
- WIN/MGT Station
- XNETMON (II)

security
- Comp. Security Checklist
- ConnectVIEW
- Dual Manager
- LAN Patrol
- SERAG
- XNETMON (II)

SMTP
- Internet Rover
- LANWatch
- mconnect
- Sniffer

SNMP
- CMU SNMP
- decaddrs
- Dual Manager
- getone
- map
- Netlabs SNMP Agent
- NETMON (III)
- NMC
- OverVIEW
- proxyd
- SNMP Kit
- Snmp Libraries
- snmpask
- snmpd (I)
- snmpd (II)
- snmplookup
- snmpperfmon
- snmppoll
- snmpquery
- snmproute
- snmpset
- snmpsrc
- snmpstat
- snmptrapd
- snmpwatch
- snmpxbar
- snmpxconn
- snmpxmon
- snmpxperf
- snmpxperfmon
- snmpxrtmetric
- Unisys NCC
- WIN/MGT Station
- xnetmon (I)
- XNETMON (II)
- xnetperfmon

sourcelib
- CMIP Library
- CMU SNMP
- HyperMIB
- Internet Rover
- LANWatch
- map
- NETMON (III)
- net_monitor
- proxyd
- SNMP Kit
- Snmp Libraries
- Snmpd (II)
- SpiderMonitor
- XNETMON (II)
- xnetperfmon

spoof
- DiG
- Internet Rover
- mconnect
- nhfsstone
- nslookup
- query
- SPIMS

standalone
- EtherMeter
- Sniffer
- SpiderMonitor

star
- LAN Patrol
- LANWatch
- map
- NETMON (III)
- proxyd
- Sniffer
- Snmp Libraries
- snmpd (II)
- XNETMON (II)
- xnetperfmon

status

- CMIP Library
- CMU SNMP
- ConnectVIEW
- DiG
- Dual Manager
- getone
- Internet Rover
- LanProbe
- mconnect
- Netlabs CMOT Agent
- Netlabs SNMP Agent
- netmon (I)
- net_monitor
- NMC
- NNStat
- NPRV
- nslookup
- osimic
- osimon
- OverVIEW
- ping
- proxyd
- sma
- SNMP Kit
- Snmp Libraries
- snmpask
- snmpd (I)
- snmpd (II)
- snmplookup
- snmpperfmon
- snmppoll
- snmpquery
- snmpstat
- snmpwatch
- snmpxbar
- snmpxconn
- snmpxmon
- snmpxperf
- snmpxperfmon
- TokenVIEW
- Unisys NCC
- WIN/MGT Station
- xnetmon (I)
- XNETMON (II)
- xnetperfmon
- xup

traffic

- ENTM
- etherfind
- EtherMeter
- EtherView
- LAN Patrol
- LanProbe
- LANWatch
- NETMON (II)
- netwatch
- Network Integrator
- nfswatch
- NMC
- NNStat
- osimon
- OSITRACE
- Sniffer
- snmpxperfmon
- SpiderMonitor
- tcpdump
- tcplogger
- TRPT
- Unisys NCC
- WIN/MGT Station

UNIX
- arp
- CMIP Library
- CMU SNMP
- decaddrs
- DiG
- Dual Manager
- etherfind
- etherhostprobe
- EtherView
- getone
- Internet Rover
- map
- mconnect
- NETMON (II)
- netstat
- Network Integrator
- net_monitor
- nfswatch
- nhfsstone
- NMC
- NNStat
- nslookup
- osilog
- osimic
- osimon
- OSITRACE
- ping
- proxyd
- query
- SERAG
- sma
- SNMP Kit
- Snmp Libraries
- snmpask
- snmpd (I)
- snmpd (II)
- snmplookup
- snmpperfmon
- snmppoll
- snmpquery
- snmproute
- snmpset
- snmpsrc
- snmpstat
- snmptrapd
- snmpwatch
- snmpxbar
- snmpxconn
- snmpxmon
- snmpxperf
- snmpxperfmon
- snmpxrtmetric
- SPIMS
- spray
- tcpdump
- tcplogger
- traceroute
- TRPT
- TTCP
- Unisys NCC
- WIN/MGT Station
- xnetmon (I)
- XNETMON (II)
- xnetperfmon

VMS
- arp
- ENTM
- netstat
- net_monitor
- NPRV
- nslookup
- ping
- Snmp Libraries
- tcpdump
- traceroute
- TTCP
- XNETMON (II)
- xnetperfmon

X
- Dual Manager
- map
- snmpxbar
- snmpxconn
- snmpxmon
- snmpxperf
- snmpxperfmon
- snmpxrtmetric
- WIN/MGT Station
- XNETMON (II)
- xnetperfmon
- xup

3. Tool Descriptions

This section is a collection of brief descriptions of tools for managing TCP/IP internets. These entries are in alphabetical order, by tool name.

The entries all follow a standard format. Immediately after the NAME of a tool are its associated KEYWORDS. Keywords are terse descriptions of the purposes or attributes of a tool. A more detailed description of a tool's purpose and characteristics is given in the ABSTRACT section. The MECHANISM section describes how a tool works. In CAVEATS, warnings about tool use are given. In BUGS, known bugs or bug-report procedures are given. LIMITATIONS describes the boundaries of a tool's capabilities. HARDWARE REQUIRED and SOFTWARE REQUIRED relate the operational environment a tool needs. Finally, in AVAILABILITY, pointers to vendors, online repositories, or other sources for a tool are given.

We deal with the problem of tool-name clashes — different tools that have the same name — by appending parenthetical roman numerals to the names. For example, BYU, MITRE, and SNMP Research each submitted a description of a tool called ''NETMON.'' These tools were independently developed, are functionally different, run in different environments, and are no more related than Richard Burton the 19th century explorer and Richard Burton the 20th century actor. BYU's tool ''NETMON'' is listed as ''NETMON (I),'' MITRE's as ''NETMON (II),'' and the tool from SNMP Research as ''NETMON (III).''

The parenthetical roman numerals reveal only the order in which the catalog editor received the tool descriptions. They should not be construed to indicate any sort of preference, priority, or rights to a tool name.

NAME

arp

KEYWORDS

routing; ethernet, IP; UNIX, VMS; free.

ABSTRACT

Arp displays and can modify the internet-to-ethernet address translations tables used by ARP, the address resolution protocol.

MECHANISM

The arp program accesses operating system memory to read the ARP data structures.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

Only the super user can modify ARP entries.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

BSD UNIX or related OS, or VMS.

AVAILABILITY

Available via anonymous FTP from uunet.uu.net, in directory bsd-sources/src/etc. Available with 4.xBSD UNIX and related operating systems. For VMS, available as part of TGV MultiNet IP software package, as well as Wollongong's WIN/TCP.

NAME

CMIP Library

KEYWORDS

alarm, control, manager, status; OSI; UNIX; free, library, sourcelib.

ABSTRACT

The CMIP Library implements the functionality of the Common Management Information Service/Protocol as in the documents ISO DP 9595-2/9596-2 of March 1988. It can act as a building block for the construction of CMIP-based agent and manager applications.

MECHANISM

The CMIP library uses ISO ROS, ACSE and ASN.1 presentation, as implemented in ISODE, to provide its service.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

The M-CREATE, M-DELETE and M-ACTION protocol primitives are not implemented in this version.

HARDWARE REQUIRED

Developed on Sun3, tested on Sun3 and VAXStation.

SOFTWARE REQUIRED

The ISODE protocol suite, BSD UNIX.

AVAILABILITY

The CMIP library and related management tools built upon it, known as OSIMIS (OSI Management Information Service), are publicly available from University College London, England via FTP and FTAM. To obtain information regarding a copy send email to gknight@ac.ucl.cs.uk or call +44 1 380 7366.

NAME

The CMU SNMP Distribution

KEYWORDS

manager, status; IP; NMS, SNMP; UNIX; free, sourcelib.

ABSTRACT

The CMU SNMP Distribution includes source code for an SNMP agent, several SNMP client applications, an ASN.1 library, and supporting documentation.

The agent compiles into about 10 KB of 68000 code. The distribution includes a full agent that runs on a Kinetics FastPath2/3/4, and is built into the KIP appletalk/ethernet gateway. The machine independent portions of this agent also run on CMU's IBM PC/AT based router.

The applications are designed to be useful in the real world. Information is collected and presented in a useful format and is suitable for everyday status monitoring. Input and output are interpreted symbolically. The tools can be used without referencing the RFCs.

MECHANISM

SNMP.

CAVEATS

None.

BUGS

None reported. Send bug reports to sw0l+snmp@andrew.cmu.edu. (''sw0l'' is ''ess double-you zero ell.'')

LIMITATIONS

None reported.

HARDWARE REQUIRED

The KIP gateway agent runs on a Kinetics FastPath2/3/4. Otherwise, no restrictions.

SOFTWARE REQUIRED

The code was written with efficiency and portability in mind. The applications compile and run on the following systems: IBM PC/RT running ACIS Release 3, Sun3/50 running SUNOS 3.5, and the DEC microVax running Ultrix 2.2. They are expected to run on any system with a Berkeley socket interface.

AVAILABILITY

This distribution is copyrighted by CMU, but may be used and sold without permission. Consult the copyright notices for further information. The distribution is available by anonymous FTP from the host lancaster.andrew.cmu.edu (128.2.13.21) as the files pub/cmu-snmp.9.tar, and pub/kip-snmp.9.tar. The former includes the libraries and the applications, and the latter is the KIP SNMP agent.

Please direct questions, comments, and bug reports to sw0l+snmp@andrew.cmu.edu. (''sw0l'' is ''ess double-you zero ell.'') If you pick up this package, please send a note to the above address, so that you may be notified of future enhancements/changes and additions to the set of applications (several are planned).

NAME

Computer Security Checklist

KEYWORDS

security; DOS.

ABSTRACT

This program consists of 858 computer security questions divided up in thirteen sections. The program presents the questions to the user and records their responses. After answering the questions in one of the thirteen sections, the user can generate a report from the questions and the user's answers. The thirteen sections are: telecommunications security, physical access security, personnel security, systems development security, security awareness and training practices, organizational and management security, data and program security, processing and operations security, ergonomics and error prevention, environmental security, and backup and recovery security.

The questions are weighted as to their importance, and the report generator can sort the questions by weight. This way the most important issues can be tackled first.

MECHANISM

The questions are displayed on the screen and the user is prompted for a single keystroke reply. When the end of one of the thirteen sections is reached, the answers are written to a disk file. The question file and the answer file are merged to create the report file.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

DOS operating system.

AVAILABILITY

A commercial product available from:

C.D., Ltd.
P.O. Box 58363
Seattle, WA 98138
(206) 243-8700

NAME
ConnectVIEW

KEYWORDS
control, manager, routing, security, status; bridge, ethernet, ring; NMS, proprietary; DOS.

ABSTRACT
The ConnectVIEW Network Management System consists of various software managers that control and manage Halley System's internets made of of ConnectLAN 100 ethernet and ConnectLAN 200 Token Ring Brouters. The management software provides an icon-based graphical network display with real-time monitoring and reporting, along with configuration, fault, performance and security management functions for managing ConnectLAN brouters. A Planning function is also provided that allows users to draw their networks.

MECHANISM
Proprietary.

CAVEATS
The ConnectVIEW software must be running under Microsoft Windows, preferably on a dedicated management station. There is, however, no degradation of LAN throughput.

BUGS
None known.

LIMITATIONS
Currently works only with Halley System's products.

HARDWARE REQUIRED
Requires a PC/AT compatible, with 640KB RAM, EGA adapter and monitor, keyboard, mouse, and ethernet adapter.

SOFTWARE REQUIRED
MSDOS 3.3 or higher. Microsoft Windows/286 version 2.1.

AVAILABILITY
Commercially available from:
Halley Systems, Inc.
2730 Orchard Parkway
San Jose, CA 95134

NAME

decaddrs, decaroute, decnroute, xnsroutes, bridgetab

KEYWORDS

manager, map, routing; bridge, DECnet; NMS, SNMP; UNIX.

ABSTRACT

These commands display private MIB information from Wellfleet systems. They retrieve and format for display values of one or several MIB variables from the Wellfleet Communications private enterprise MIB, using the SNMP (RFC1098). In particular these tools are used to examine the non-IP modules (DECnet, XNS, and Bridging) of a Wellfleet system.

Decaddrs displays the DECnet configuration of a Wellfleet system acting as a DECnet router, showing the static parameters associated with each DECnet interface. Decaroute and decnroute display the DECnet inter-area and intra-area routing tables (that is area routes and node routes). Xnsroutes displays routes known to a Wellfleet system acting as an XNS router. Bridgetab displays the bridge forwarding table with the disposition of traffic arriving from or directed to each station known to the Wellfleet bridge module. All these commands take an IP address as the argument and can specify an SNMP community for the retrieval. One SNMP query is performed for each row of the table. Note that the Wellfleet system must be operating as an IP router for the SNMP to be accessible.

MECHANISM

Management information is exchanged by use of SNMP.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

Distributed and supported for Sun 3 systems.

SOFTWARE REQUIRED

Distributed and supported for SunOS 3.5 and 4.x.

AVAILABILITY

Commercial product of:

Wellfleet Communications, Inc.
12 DeAngelo Drive
Bedford, MA 01730-2204
(617) 275-2400

NAME
DiG

KEYWORDS
status; DNS; spoof; UNIX; free.

ABSTRACT
DiG (domain information groper), is a command line tool which queries DNS servers in either an interactive or a batch mode. It was developed to be more convenient/flexible than nslookup for gathering performance data and testing DNS servers.

MECHANISM
Dig is built on a slightly modified version of the bind resolver (release 4.8).

CAVEATS
none.

BUGS
None known.

LIMITATIONS
None reported.

HARDWARE REQUIRED
No restrictions.

SOFTWARE REQUIRED
BSD UNIX.

AVAILABILITY
DiG is available via anonymous FTP from venera.isi.edu in pub/dig.1.0.tar.Z.

NAME

Dual Manager

KEYWORDS

alarm, control, manager, map, security, status; IP, OSI; NMS, SNMP, X; UNIX; library.

ABSTRACT

Netlabs' Dual Manager provides management of TCP/IP networks using both SNMP and CMOT protocols. Such management can be initiated either through the X-Windows user interface (both Motif and Openlook), or through OSI Network Management (CMIP) commands. The Dual Manager provides for configuration, fault, security and performance management. It provides extensive map management features, including scanned maps in the background. It provides simple mechanisms to extend the MIB and assign specific lists of objects to specific network elements, thereby providing for the management of all vendors' specific MIB extensions. It provides an optional relational DBMS for storing and retrieving MIB and alarm information. Finally, the Dual Manager is an open platform, in that it provides several Application Programming Interfaces (APIs) for users to extend the functionality of the Dual Manager.

The Dual Manager is expected to work as a TCP/IP ‘‘branch manager’’ under DEC's EMA, AT&T's UNMA and other OSI-conformant enterprise management architectures.

MECHANISM

The Netlabs Dual Manager supports the control and monitoring of network resources by use of both CMOT and SNMP message exchanges.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

Runs on Sun/3 and Sun/4s.

SOFTWARE REQUIRED

Available on System V or SCO Open Desktop environments. Uses X-Windows for the user interface.

AVAILABILITY

Commercially available from:

Netlabs Inc
11693 Chenault Street Ste 348
Los Angeles CA 90049
(213) 476-4070
lam@netlabs.com (Anne Lam)

NAME

ENTM — Ethernet Traffic Monitor

KEYWORDS

traffic; ethernet, IP; eavesdrop; VMS; free.

ABSTRACT

ENTM is a screen-oriented utility that runs under VAX/VMS. It monitors local ethernet traffic and displays either a real time or cumulative, histogram showing a percent breakdown of traffic by ethernet protocol type. The information in the display can be reported based on packet count or byte count. The percent of broadcast, multicast and approximate lost packets is reported as well. The screen display is updated every three seconds. Additionally, a real time, sliding history window may be displayed showing ethernet traffic patterns for the last five minutes.

ENTM can also report IP traffic statistics by packet count or byte count. The IP histograms reflect information collected at the TCP and UDP port level, including ICMP type/code combinations. Both the ethernet and IP histograms may be sorted by ASCII protocol/port name or by percent-value. All screen displays can be saved in a file for printing later.

MECHANISM

This utility simply places the ethernet controller in promiscuous mode and monitors the local area network traffic. It preallocates 10 receive buffers and attempts to keep 22 reads pending on the ethernet device.

CAVEATS

Placing the ethernet controller in promiscuous mode may severly slow down a VAX system. Depending on the speed of the VAX system and the amount of traffic on the local ethernet, a large amount of CPU time may be spent on the Interrupt Stack. Running this code on any production system during operational hours is discouraged.

BUGS

Due to a bug in the VAX/VMS ethernet/802 device driver, IEEE 802 format packets may not always be detected. A simple test is performed to ‘‘guess’’ which packets are in IEEE 802 format (DSAP equal to SSAP). Thus, some DSAP/SSAP pairs may be reported as an ethernet type, while valid ethernet types may be reported as IEEE 802 packets.

In some hardware configurations, placing an ethernet controller in promiscuous mode with automatic-restart enabled will hang the controller. Our VAX 8650 hangs running this code, while our uVAX IIs and uVAX IIIs do not.

Please report any additional bugs to the author at:

Allen Sturtevant
National Magnetic Fusion Energy Computer Center
Lawrence Livermore National Laboratory
P.O. Box 808; L-561
Livermore, CA 94550
Phone : (415) 422-8266
E-Mail: sturtevant@ccc.nmfecc.gov

LIMITATIONS

The user is required to have PHY_IO, TMPMBX and NETMBX privileges. When activated, the program first checks that the user process as enough quotas remaining (BYTLM, BIOLM, ASTLM and PAGFLQUO) to successfully run the program without entering into an involuntary wait state. Some quotas require a fairly generous setting.

The contents of IEEE 802 packets are not examined. Only the presence of IEEE 802 packets on the wire is reported.

The count of lost packets is approximated. If, after each read completes on the ethernet device, the utility detects that it has no reads pending on that device, the lost packet counter is incremented by one.

When the total number of bytes processed exceeds 7fffffff hex, all counters are automatically reset to zero.

HARDWARE REQUIRED

A DEC ethernet controller.

SOFTWARE REQUIRED

VAX/VMS version V5.1+.

AVAILABILITY

For executables only, FTP to the ANONYMOUS account (password GUEST) on CCC.NMFECC.GOV and GET the following files:

[ANONYMOUS.PROGRAMS.ENTM]ENTM.DOC	(ASCII text)
[ANONYMOUS.PROGRAMS.ENTM]ENTM.EXE	(binary)
[ANONYMOUS.PROGRAMS.ENTM]EN_TYPES.DAT	(ASCII text)
[ANONYMOUS.PROGRAMS.ENTM]IP_TYPES.DAT	(ASCII text)

NAME

etherfind

KEYWORDS

traffic; ethernet, IP, NFS; eavesdrop; UNIX.

ABSTRACT

Etherfind examines the packets that traverse a network interface, and outputs a text file describing the traffic. In the file, a single line of text describes a single packet: it contains values such as protocol type, length, source, and destination. Etherfind can print out all packet traffic on the ethernet, or traffic for the local host. Further packet filtering can be done on the basis of protocol: IP, ARP, RARP, ICMP, UDP, ND, TCP, and filtering can also be done based on the source, destination addresses as well as TCP and UDP port numbers.

MECHANISM

In usual operations, and by default, etherfind puts the interface in promiscuous mode. In 4.3BSD UNIX and related OSs, it uses a Network Interface Tap (NIT) to obtain a copy of traffic on an ethernet interface.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

Minimal protocol information is printed. Can only be run by the super user. The syntax is painful.

HARDWARE REQUIRED

Ethernet.

SOFTWARE REQUIRED

SunOS.

AVAILABILITY

Executable included in Sun OS ‘‘Networking Tools and Programs’’ software installation option.

NAME

etherhostprobe

KEYWORDS

map, routing; ethernet, IP; ping; UNIX; free.

ABSTRACT

Output list of hosts on an ethernet that respond to IP ARP. Produces a list in the following format:

08:00:20:01:96:62	128.18.4.114	apptek4
08:00:20:00:02:fe	128.18.4.115	apptek5
08:00:20:00:57:6a	128.18.4.116	apptek6
08:00:20:00:65:34	128.18.4.117	apptek7
08:00:20:06:58:6f	128.18.4.118	apptek8
08:00:20:00:03:4f	128.18.4.119	apptek9

The first column is the ethernet address, the second the IP address, and the third is the hostname (which is omitted if the name could not be found via gethostbyaddr). A starting and ending IP address may be specified on the command line, which will limit the search.

MECHANISM

Etherhostprobe sends a UDP packet to the ‘‘echo’’ port, then looks in the kernel’s ARP cache for the corresponding address entry. Explicit response (or lack of same) to the UDP packet is ignored. The cache will be checked up to four times at one-quarter-second intervals. Note that this allows the program to be run by a user with no special privileges.

CAVEATS

Etherhostprobe will fill the kernel’s ARP cache with possibly useless entries, possibly causing delays to programs foolishly attempting to accomplish real work.

Etherhostprobe causes -lots- of ARPs to be generated, possibly fooling network monitoring software (or people) into concluding that something is horribly broken.

Etherhostprobe spends up to one second looking for each possible address. Thus, exhaustively searching a class-C network will take about four minutes, and exhaustively searching a class-B network will take about 18 hours. Exhaustively searching a class-A network will take the better part of a year, so don’t even think about it.

Etherhostprobe will be fooled by gateways that implement proxy ARP; every possible address on the proxy-ARPed subnet will be listed with the gateway’s ethernet address.

BUGS

None known.

LIMITATIONS

If a given machine is not running IP ARP at the time that it is probed, it will be considered nonexistent. In particular, if a given machine is down at the time that it is probed . . .

All hosts being probed must be on the same (possibly bridged) ethernet.

HARDWARE REQUIRED

No restrictions, but see below.

SOFTWARE REQUIRED

Runs on SunOS 3.5, and possibly elsewhere. The major non-standard portion of code is ‘‘tx_arp.c’’, which reads the kernel’s ARP cache.

AVAILABILITY

Copyrighted, but freely distributed. Available via anonymous FTP from spam.itstd.sri.com (128.18.10.1). From pub directory, file EHP.1 for etherhostprobe, and files IPF.1 and IPF.2 for ipForwarding.

NAME

EtherMeter (tm), model LANB/150

KEYWORDS

alarm, map, traffic; ethernet; NMS, proprietary; standalone.

ABSTRACT

The Network Applications Technology (NAT) EtherMeter product is a dedicated ethernet traffic monitor that provides statistics on the ethernet segment to which it is attached. The EtherMeter reports three major kinds of statistics. For good packets, it reports the total number of good packets seen on the segment, the number of multicast and broadcast packets, and the total number of bytes in all packets seen. For packets with errors, it reports the number of CRC errors, short packets, oversize packets, and alignment errors. It also reports the distribution of packet by type, and the number of protocols seen on the segment. A count of transmit collisions is reported. Peak and current ethernet utilization rates are also reported, etc. Alarms can be set for utilization rate, packet rate, total error count, and delta error.

The EtherMeter reports the statistics to a Network Management Station (NMS), also available from NAT, via IP/UDP datagrams, so that the meters can be monitored through routers. The NMS displays graphical and/or textual information, and EtherMeter icons turn colors to indicate status. Alarms can be set, and if the levels are exceeded an audible alarm is generated on the NMS, and the EtherMeter icon changes from green to yellow on the network map.

MECHANISM

The EtherMeter is a self-contained board that can either be plugged into a PC/AT bus for power or installed in a small stand-alone enclosure. The board can be obtained with either a 10BASE5 thick ethernet transceiver cable connector, or a 10BASE2 thin ethernet BNC connector.

CAVEATS

The EtherMeter is primarily a passive device whose only impact on the network will come from the monitoring packets sent to the NMS. The EtherMeter is assigned an IP address for communication with the NMS.

BUGS

None known.

LIMITATIONS

Proprietary protocol currently in use. The company has stated its intention to develop SNMP for the EtherMeter product in the first half of 1990. Currently the NMS does not keep log files. This limitation is acknowledged, and plans are underway to add ASCII log file capability to the NMS.

HARDWARE REQUIRED

An EtherMeter board and a PC/AT bus to plug it into, or a stand-alone enclosure with power supply (available from NAT). A Network Management Station and its software is required as well, to fully interact with the EtherMeter devices.

SOFTWARE REQUIRED

The EtherMeter software is included in ROM on the device. The NMS software is bundled in with the NMS hardware.

AVAILABILITY

The EtherMeter device, stand-alone enclosure, and Network Management Station, are available commercially from:

Network Application Technology, Inc.
21040 Homestead Road
Cupertino, California 95014
Phone: (408) 733-4530
Fax: (408) 733-6478

NAME
EtherView(tm)

KEYWORDS
traffic; ethernet, IP, NFS; eavesdrop; UNIX.

ABSTRACT
EtherView is a network monitoring tool which runs on Sun workstations and allows you to monitor your heterogeneous internet network. It monitors all systems on the ethernet. It has three primary functions:

Load Profile: It allows users to monitor the load on the ethernet over extended periods of time. The network administrator can use it to characterize load generated by a node on the network, determine which systems and applications generate how much of the load and how that load fluctuates over long periods of time.

NFS Profile: It allows the network administrator to determine the load on NFS servers, the average response time NFS servers and the mix of NFS load on each of the servers. Users can use the data to benchmark different NFS servers, determine which servers are overloaded, deduce the number of clients that each server can support and evaluate the effectiveness of NFS accelerators.

Protocol Analyzer: Users can capture packets based on source, destination, application, protocol, bit pattern, packet size or a boolean filtering expression. It provides all standard features such as configurable buffer size, packet slicing and bit pattern based triggering criterion. It does automatic disassembly of NFS, TCP, UDP, IP, ICMP, ARP and RARP packets. Packets can be examined in any combination of summary, hex or detail format.

MECHANISM
EtherView uses the Sun's NIT interface to turn the ethernet interface into promiscuous mode to capture packets. A high level process manages the interface and a low level process does the actual capturing and filtering. Shared memory is used to communicate between the two processes.

BUGS
None known.

LIMITATIONS
Because of limitations in Sun's NIT interface, EtherView will not capture packets originating from the system where it is run.

EtherView requires super-user privileges on the system where it is run.

HARDWARE REQUIRED
EtherView runs on all models of Sun-3, Sun-4 and Sun-386i.

SOFTWARE REQUIRED

Sun-3	- SunOS 4.0.3. (SunOS 4.0 with NIT fixes).
Sun-4	- SunOS 4.0.
Sun-386i	- SunOS 4.0.

Runs under SunView.
Will run under X Windows in future.

AVAILABILITY

EtherView is copyrighted, commercial product of:

Matrix Computer Systems, Inc.
7 1/2 Harris Road
Nashua, NH 03062

Tel: (603) 888-7790
email: ...uunet!matrix!eview

NAME
getone, getmany, getroute, getarp, getaddr, getif, getid.

KEYWORDS
manager, routing, status; IP; NMS, SNMP; UNIX.

ABSTRACT
These commands retrieve and format for display values of one or several MIB variables (RFC1066) using the SNMP (RFC1098). Getone and getmany retrieve arbitrary MIB variables; getroute, getarp, getaddr, and getif retrieve and display tabular information (routing tables, ARP table, interface configuration, etc.), and getid retrieves and displays system name, identification and boot time.

Getone <target> <mibvariable> retrieves and displays the value of the designated MIB variable from the specified target system. The SNMP community name to be used for the retrieval can also be specified. Getmany works similarly for groups of MIB variables rather than individual values. The name of each variable, its value and its data type is displayed. Getroute returns information from the ipRoutingTable MIB structure, displaying the retrieved information in an accessible format. Getarp behaves similarly for the address translation table; getaddr for the ipAddressTable; and getif displays information from the interfaces table, supplemented with information from the ipAddressTable. Getid displays the system name, identification, ipForwarding state, and the boot time and date. All take a system name or IP address as an argument and can specify an SNMP community for the retrieval. One SNMP query is performed for each row of the table.

MECHANISM
Queries SNMP agent(s).

CAVEATS
None.

BUGS
None known.

LIMITATIONS
None reported.

HARDWARE REQUIRED
Distributed and supported for Sun 3 systems.

SOFTWARE REQUIRED
Distributed and supported for SunOS 3.5 and 4.x.

AVAILABILITY
Commercial product of:
Wellfleet Communications, Inc.
12 DeAngelo Drive
Bedford, MA 01730-2204
(617) 275-2400

NAME
: hammer & anvil

KEYWORDS
: benchmark, generator; IP; DOS; free.

ABSTRACT
: Hammer and anvil are the benchmarking programs for IP routers. Using these tools, gateways have been tested for per-packet delay, router-generated traffic overhead, maximum sustained throughput, etc.

MECHANISM
: Tests are performed on a gateway in an isolated testbed. Hammer generates packets at controlled rates. It can set the length and interpacket interval of a packet stream. Anvil counts packet arrivals.

CAVEATS
: Hammer should not be run on a live network.

BUGS
: None reported.

LIMITATIONS
: Early versions of hammer could not produce inter-packet intervals shorter than 55 usec.

HARDWARE REQUIRED
: Hammer runs on a PC/AT or compatible, and anvil requires a PC or clone. Both use a Micom Interlan NI5210 for LAN interface.

SOFTWARE REQUIRED
: MS-DOS.

AVAILABILITY
: Hammer and anvil are copyrighted, though free. Copies are available from pub/eutil on husc6.harvard.edu.

NAME

hopcheck

KEYWORDS

routing; IP; ping; DOS; free.

ABSTRACT

Hopcheck is a tool that lists the gateways traversed by packets sent from the hopcheck-resident PC to a destination. Hopcheck uses the same mechanism as traceroute but is for use on IBM PC compatibles that have ethernet connections. Hopcheck is part of a larger TCP/IP package that is known as ka9q that is for use with packet radio. Ka9q can coexist on a PC with other TCP/IP packages such as FTP Inc's PC/TCP, but must be used independently of other packages. Ka9q was written by Phil Karn. Hopcheck was added by Katie Stevens, dkstevens@ucdavis.edu. Unlike traceroute, which requires a UNIX kernel mod, hopcheck will run on the standard, unmodified ka9q release.

MECHANISM

See the description in traceroute.

CAVEATS

See the description in traceroute.

BUGS

None known.

LIMITATIONS

Host table required. Does not work with domain name server or with IP address as the argument. This is mainly an inconvenience.

HARDWARE REQUIRED

IBM PC compatible with ethernet network interface card, though does not work with 3Com 505 board.

SOFTWARE REQUIRED

DOS.

AVAILABILITY

Free. On deposit at the National Center for Atmospheric Research. For access from UNIX, available via anonymous FTP from windom.ucar.edu, in directory ''etc,'' as hopcheck.tar.Z. For access directly from a PC, fetch nethop.exe and readme.hop; nethop.exe is executable. Also available via anonymous FTP at ucdavis.edu, in the nethopexe or nethopsrc suite of files in directory ''dist.''

NAME

HyperMIB

KEYWORDS

reference; Macintosh; free, sourcelib.

ABSTRACT

HyperMIB is a hypertext presentation of the MIB (RFC1066). The tree structure of the MIB is presented graphically, and the user traverses the tree by selecting branches of the tree. When the MIB variables are displayed, selecting them causes a text window to appear and show the definition of that variable (using the actual text of the MIB document).

MECHANISM

The Apple Macintosh HyperCard utility is used. The actual text of the MIB document is read into scrollable text windows, and a string search is done on the variable selected. A person familiar with HyperCard programming could modify the program to suit their needs (such as to add the definitions for their company's private space).

CAVEATS

None.

BUGS

None known.

LIMITATIONS

This program only gives the definition of the MIB variables. It cannot poll a node to find the value of the variables.

HARDWARE REQUIRED

Apple Macintosh computer with at least 1MByte of RAM.

SOFTWARE REQUIRED

Apple Macintosh operating system and HyperCard.

AVAILABILITY

This software may be copied and given away without charge. The files are available by anonymous FTP on CCC.NMFECC.GOV. The files are:

[Anonymous.programs.HyperMIB]Hyper_MIB.help	(ASCII text)
[Anonymous.programs.HyperMIB]Hyper.MIB	(binary)
[Anonymous.programs.HyperMIB]MIB.tree	(binary)

The software is also available for a nominal fee from:

National Energy Software Center
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439
(312) 972-7250

NAME

Internet Rover

KEYWORDS

status; IP, SMTP; curses, ping, spoof; UNIX; free, sourcelib.

ABSTRACT

Internet Rover is a prototype network monitor that uses multiple protocol ‘‘modules’’ to test network functionality. This package consists of two primary pieces of code: the data collector and the problem display.

There is one data collector that performs a series of network tests, and maintains a list of problems with the network. There can be many display processes all displaying the current list of problems which is useful in a multi-operator NOC.

The display task uses curses, allowing many terminal types to display the problem file either locally or from a remote site. Full source is provided. The data collector is easily configured and extensible. Contributions such as additional protocol modules, and shell script extensions are welcome.

MECHANISM

A configuration file contains a list of nodes, addresses, NodeUp? protocol test (ping in most cases), and a list of further tests to be performed if the node is in fact up. Modules are included to test TELNET, FTP, and SMTP. If the configuration contains a test that isn’t recognized, a generic test is assumed, and a filename is checked for existence. This way users can create scripts that create a file if there is a problem, and the data collector simply checks the existence of that file to determine if there is problem.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

This tools does not yet have the capability to perform actions based on the result of the test. Rather, it is intended for a multi-operator environment, and simply displays a list of what is wrong with the net.

HARDWARE REQUIRED

This software is known to run on Suns and IBM RTs.

SOFTWARE REQUIRED

Curses, 4.xBSD UNIX socket programming libraries, BSD ping.

AVAILABILITY

Full source available via anonymous FTP from merit.edu (35.1.1.42) in the ~ftp/pub/inetrover directory. Source and executables are public domain and can be freely distributed for non-commercial use. This package is unsupported, but bug reports and fixes may be sent to: wbn@merit.edu.

NAME
LAN Patrol

KEYWORDS
security, traffic; ethernet, star; eavesdrop; DOS.

ABSTRACT
LAN Patrol is a full-featured network analyzer that provides essential information for effective fault and performance management. It allows network managers to easily monitor user activity, find traffic overloads, plan for growth, test cable, uncover intruders, balance network services, and so on. LAN Patrol uses state of the art data collection techniques to monitor all activity on a network, giving an accurate picture of how it is performing.

LAN Patrol's reports can be saved as ASCII files to disk, and imported into spreadsheet or database programs for further analysis.

MECHANISM
The LAN Patrol interface driver programs a standard interface card to capture all traffic on a network segment. The driver operates from the background of a standard PC, maintaining statistics for each station on the network. The information can be viewed on the PC's screen, or as a user-defined report output either to file or printer.

CAVEATS
None. Normal operation is completely passive, making LAN Patrol transparent to the network.

BUGS
None known.

LIMITATIONS
LAN Patrol can monitor up to 10,000 packets/sec on an AT class PC, and is limited to monitoring a maximum of 1024 stations for intervals of up to 30 days.

Because LAN Patrol operates at the physical level, it will only see traffic for the segment on which it is installed; it cannot see traffic across bridges.

HARDWARE REQUIRED
Computer: IBM PC/XT/AT, PS/2 Model 30, or compatible. Requires 512K memory and a hard drive or double-sided disk drive.

Display: Color or monochrome text. Color display allows color-coding of traffic information.

Ethernet, StarLAN, LattisNet, or StarLAN 10 network interface card.

SOFTWARE REQUIRED
PC DOS, MS-DOS version 3.1 or greater.

AVAILABILITY
LAN Patrol many be purchased through network dealers, or directly from:

Legend Software, Inc.
Phone: (201) 227-8771
FAX: (201) 906-1151

NAME

LanProbe — the HP 4990S LanProbe Distributed Analysis System.

KEYWORDS

alarm, manager, map, status, traffic; ethernet; eavesdrop, NMS; proprietary.

ABSTRACT

The LanProbe distributed monitoring system performs remote and local monitoring of ethernet LANs in a protocol and vendor independent manner.

LanProbe discovers each active node on a segment and displays it on a map with its adapter card vendor name, ethernet address, and IP address. Additional information about the nodes, such as equipment type and physical location can be entered in to the data base by the user.

When the NodeLocator option is used, data on the actual location of nodes is automatically entered and the map becomes an accurate representation of the physical layout of the segment. Thereafter when a new node is installed and becomes active, or when a node is moved or becomes inactive, the change is detected and shown on the map in real time. The system also provides the network manager with precise cable fault information displayed on the map.

Traffic statistics are gathered and displayed and can be exported in (comma delimited) CSV format for further analysis. Alerts can be set on user defined thresholds.

Trace provides a remote protocol analyzer capability with decodes for common protocols.

Significant events (like power failure, cable breaks, new node on network, broadcast IP source address seen, etc.) are tracked in a log that is uploaded to ProbeView periodically.

ProbeView generates reports that can be manipulated by MSDOS based word processors, spreadsheets, and DBMS.

MECHANISM

The system consists of one or more LanProbe segment monitors and ProbeView software running under Microsoft Windows. The LanProbe segment monitor attaches to the end of an ethernet segment and monitors all traffic. Attachment can be direct to a thin or thick coax cable, or via an external transceiver to fiber optic or twisted pair cabling. Network data relating to the segment is transferred to a workstation running ProbeView via RS-232, ethernet, or a modem connection.

ProbeView software, which runs on a PC/AT class workstation, presents network information in graphical displays.

The HP4992A NodeLocator option attaches to the opposite end of the cable from the HP4991A LanProbe segment monitor. It automatically locates the position of nodes on the ethernet networks using coaxial cabling schemes.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

HP 4991A LanProbe segment monitor
HP 4992A NodeLocator (for optional capabilities)
80386 based PC capable of running MS-Windows

SOFTWARE REQUIRED

HP 4990A ProbeView

MSDOS 3.0 or higher and Microsoft Windows/286 2.1.

AVAILABILITY

A commercial product available from:

Hewlett-Packard Company
P.O. Box 10301,
Palo Alto, CA 94303-0890

NAME
LANWatch

KEYWORDS
alarm, analyzer, traffic; CHAOS, DECnet, DNS, ethernet, IP, OSI, ring, SMTP, star; eavesdrop; DOS; library, sourcelib.

ABSTRACT
LANWatch 2.0 is an inexpensive, powerful and flexible network analyzer that runs under DOS on personal computers and requires no hardware modifications to either the host or the network. LANWatch is an invaluable tool for installing, troubleshooting, and monitoring local area networks, and for developing and debugging new protocols. Network managers using LANWatch can inspect network traffic patterns and packet errors to isolate performance problems and bottlenecks. Protocol developers can use LANWatch to inspect and verify proper protocol handling. Since LANWatch is a software-only package which installs easily in existing PCs, network technicians and field service engineers can carry LANWatch in their briefcase for convenient network analysis at remote sites.

LANWatch has two operating modes: Display and Examine. In Display Mode, LANWatch traces network traffic by displaying captured packets in real time. Examine Mode allows you to scroll back through stored packets to inspect them in detail. To select a subset of packets for display, storage or retrieval, there is an extensive set of built-in filters. Using filters, LANWatch collects only packets of interest, saving the user from having to sort through all network traffic to isolate specific packets. The built-in filters include alarm, trigger, capture, load, save and search. They can be controlled separately to match on source or destination address, protocol, or packet contents at the hardware and transport layers. LANWatch also includes sufficient source code so users can modify the existing filters and parsers or add new ones.

The LANWatch distribution includes executables and source for several post-processors: a TCP protocol analyzer, a node-by-node traffic analyzer and a dump file listing tool.

MECHANISM
Uses many common PC network interfaces by placing them in promiscuous mode and capturing traffic.

CAVEATS
Most PC network interfaces will not capture 100% of the traffic on a fully-loaded network (primarily missing back-to-back packets).

BUGS
None known.

LIMITATIONS
LANWatch can't analyze what it doesn't see (see Caveats).

HARDWARE REQUIRED
LANWatch requires a PC or PS/2 with a supported network interface card.

SOFTWARE REQUIRED
LANWatch runs in DOS. Modification of the supplied source code or creation of additional filters and parsers requires Microsoft C 5.1

AVAILABILITY
LANWatch is commercially available from FTP Software, Incorporated, 26 Princess Street, Wakefield, MA, 01880 (617 246-0900).

NAME

map — Interactive Network Map

KEYWORDS

manager, map; CHAOS, ethernet, IP, ring, star; NMS, ping, SNMP, X; UNIX; free, sourcelib.

ABSTRACT

Map draws a map of network connectivity and allows interactive examination of information about various components including whether hosts can be reached over the network.

The program is supplied with complete source and is written in a modular fashion to make addition of different protocols stacks, displays, or hardcopy devices relatively easy. This is one of the reasons why the initial version supports at least two of each. Contributions of additional drivers in any of these areas will be welcome as well as porting to additional platforms.

MECHANISM

Net components are pinged by use of ICMP echo and, optionally, CHAOS status requests and SNMP ‘‘gets.’’ The program initializes itself from static data stored in the file system and therefore does not need to access the network in order to get running (unless the static files are network mounted).

CAVEATS

As of publication, the tool is in beta release.

BUGS

Several minor nits, documented in distribution files. Bug discoveries should be reported by email to Bug-Map@LCS.MIT.Edu.

LIMITATIONS

See distribution file for an indepth discussion of system capabilities and potential.

HARDWARE REQUIRED

An X display is needed for interactive display of the map, non-graphical interaction is available in non-display mode. For hardcopy output a PostScript or Tektronix 4692 printer is required.

SOFTWARE REQUIRED

BSD UNIX or related OS. IP/ICMP is required; CHAOS/STATUS and SNMP can be used but are optional. X-Windows is required for interactive display of the map.

AVAILABILITY

As of publication, map is in beta release. To be added to the email forum that discusses the software, or to obtain individual files or instructions on getting the full current release, send a request to:

MAP-Request@LCS.MIT.Edu.

The program is Copyright MIT. It is available via anonymous FTP with a license making it free to use and distribute for non-commercial purposes.

NAME
mconnect

KEYWORDS
status; SMTP; spoof; UNIX.

ABSTRACT
Mconnect allows an interactive session with a remote mailer. Mail delivery problems can be diagnosed by connecting to the remote mailer and issuing SMTP commands directly.

MECHANISM
Opens a TCP connection to remote SMTP on port 25. Provides local line buffering and editing, which is the distinction between mconnect and a TELNET to port 25.

CAVEATS
None.

BUGS
None known.

LIMITATIONS
Mconnect is not a large improvement over using a TELNET connection to port 25.

HARDWARE REQUIRED
No restrictions.

SOFTWARE REQUIRED
BSD UNIX or related OS.

AVAILABILITY
Available with 4.xBSD UNIX and related operating systems.

NAME

Netlabs CMOT Agent

KEYWORDS

manager, status; IP, OSI; NMS.

ABSTRACT

Netlabs' CMOT code debuted in Interop 89. The CMOT code comes with an Extensible MIB, which allows users to add new MIB variables. The code currently supports all the MIB variables in RFC 1095 via the data types in RFC 1065, as well as the emerging MIB-II, which is currently in experimental stage. The CMOT has been benchmarked at 100 Management Operations per Second (MOPS) for a 1-MIPS machine.

MECHANISM

The Netlabs CMOT agent supports the control and monitoring of network resources by use of CMOT message exchanges.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

Portable to most hardware.

SOFTWARE REQUIRED

Portable to most operating systems.

AVAILABILITY

Commercially available from:

Netlabs Inc
11693 Chenault Street Ste 348
Los Angeles CA 90049
(213) 476-4070
lam@netlabs.com (Anne Lam)

NAME

Netlabs SNMP Agent.

KEYWORDS

manager, status; IP; NMS, SNMP.

ABSTRACT

Netlabs' SNMP code debuted in Interop 89, where it showed interoperation of the code with several implementations on the show floor. The SNMP code comes with an Extensible MIB, which allows users to add new MIB variables. The code currently supports all the MIB variables in RFC 1066 via the data types in RFC 1065, as well as the emerging MIB-II, which is currently in experimental stage. The SNMP has been benchmarked at 200 Management Operations per Second (MOPS) for a 1-MIPS machine.

MECHANISM

The Netlabs SNMP agent supports the control and monitoring of network resources by use of SNMP message exchanges.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

Portable to most hardware.

SOFTWARE REQUIRED

Portable to most operating systems.

AVAILABILITY

Commercially available from:

Netlabs Inc
11693 Chenault Street Ste 348
Los Angeles CA 90049
(213) 476-4070
lam@netlabs.com (Anne Lam)

NAME

netmon

KEYWORDS

status; DNS, IP; ping; DOS; free.

ABSTRACT

Netmon is a DOS-based program that pings hosts on a monitored list at user-specified intervals. In addition, a user may optionally ping hosts not on the list.

Netmon also performs domain lookups. Furthermore, a user may build and send a domain query to any desired DNS server.

MECHANISM

The tool works by using the echo service feature of ICMP. It reports if it receives an incorrect response or no response.

CAVEATS

Depending on the frequency of pinging and the number of hosts pinged, netmon could create a high volume of traffic.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

A PC, and a Western Digital WD8003 interface card (or any other card for which there is a packet driver for FTP Software Inc.'s PC/TCP kernel). Both monochrome and color displays are supported, though color is recommended.

SOFTWARE REQUIRED

DOS operating system, and the PC/TCP Kernel by FTP Software, Inc.

AVAILABILITY

The BYU modified version is available for anonymous FTP from Dcsprod.byu.edu, in directory ''programs.'' It can be freely distributed for non-commercial use.

NAME

NETMON and iptrace

KEYWORDS

traffic; IP; eavesdrop; UNIX; free.

ABSTRACT

NETMON is a facility to enable communication of networking events from the BSD UNIX operating system to a user-level network monitoring or management program. Iptrace is a program interfacing to NETMON which logs TCP-IP traffic for performance measurement and gateway monitoring. It is easy to build other NETMON-based tools using iptrace as a model.

NETMON resides in the 4.3BSD UNIX kernel. It is independent of hardware-specific code in UNIX. It is transparent to protocol and network type, having no internal assumptions about the network protocols being recorded. It is installed in BSD-like kernels by adding a standard function call (probe) to a few points in the input and output routines of the protocols to be logged.

NETMON is analogous to Sun Microsystems' NIT, but the interface tap function is extended by recording more context information. Aside from the timestamp, the choice of information recorded is up to the installer of the probes. The NETMON probes added to the BSD IP code supplied with the distribution include as context: input and output queue lengths, identification of the network interface, and event codes labeling packet discards. (The NETMON distribution is geared towards measuring the performance of BSD networking protocols in an IP gateway).

NETMON is designed so that it can reside within the monitored system with minimal interference to the network processing. The estimated and measured overhead is around five percent of packet processing.

The user-level tool ''iptrace'' is provided with NETMON. This program logs IP traffic, either at IP-level only, or as it passes through the network interface drivers as well. As a separate function, iptrace produces a host traffic matrix output. Its third type of output is abbreviated sampling, in which only a pre-set number of packets from each new host pair is logged. The three output types are configured dynamically, in any combination.

OSITRACE, another logging tool with a NETMON interface, is available separately (and documented in a separate entry in this catalog).

MECHANISM

Access to the information logged by NETMON is through a UNIX special file, /dev/netmon. User reads are blocked until the buffer reaches a configurable level of fullness.

Several other parameters of NETMON can be tuned at compile time. A diagnostic program, netmonstat, is included in the distribution.

CAVEATS

None.

BUGS

Bug reports and questions should be addressed to:

ie-tools@gateway.mitre.org

Requests to join this mailing list:

ie-tools-request@gateway.mitre.org

Questions and suggestions can also be directed to:

Allison Mankin (703)883-7907
mankin@gateway.mitre.org

LIMITATIONS

A NETMON interface for tcpdump and other UNIX protocol analyzers is not included, but it is simple to write. NETMON probes for a promiscuous ethernet interface are similarly not included.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

BSD UNIX-like network protocols or the ability to install the BSD publicly available network protocols in the system to be monitored.

AVAILABILITY

The NETMON distribution is available by anonymous FTP in pub/netmon.tar or pub/netmon.tar.Z from aelred-3.ie.org. A short user's and installation guide, NETMON.doc, is available in the same location. The NETMON distribution is provided ''as is'' and requires retention of a copyright text in code derived from it. It is copyrighted by the MITRE-Washington Networking Center.

NAME

NETMON — an SNMP-based network management tool from SNMP Research.

KEYWORDS

alarm, control, manager, map, routing; DECnet, ethernet, IP, OSI, ring, star; NMS, SNMP; DOS; sourcelib.

ABSTRACT

The NETMON application implements a network management station based on a low-cost DOS-based platform. It can be successfully used with many types of networks, including both wide area networks and those based on various LAN media. NETMON has been used with multiprotocol devices including those which support TCP/IP, DECnet, and OSI protocols. The fault management tool displays the map of the network configuration with current node and link state indicated in one of several colors. Alarms may be enabled to alert the operator of events occurring in the network. Events are logged to disk. The NETMON application comes complete with source code including a powerful set of portable libraries for generating and parsing SNMP messages. Output data from NETMON may be transferred via flat files for additional report generation by a variety of statistical packages.

MECHANISM

The NETMON application is based on the Simple Network Management Protocol (SNMP). Polling is performed via the powerful SNMP get-next operator and the SNMP get operator. Trap directed polling is used to regulate the focus and intensity of the polling.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

The monitored and managed nodes must implement the SNMP over UDP per RFC 1098 or must be reachable via a proxy agent.

HARDWARE REQUIRED

The minimum system is a IBM Personal Computer (4.77 MHz) with DOS 3.0 or later, an Enhanced Graphics Adapter, Enhanced Graphics Monitor, a single 360 Kbyte floppy drive, and an ethernet adapter. However, most users will find a hard disk to be helpful for storing network history and will be less impatient with a faster CPU.

SOFTWARE REQUIRED

DOS 3.0 or later and TCP/IP software from one of several sources.

AVAILABILITY

This is a commercial product available under license from:

SNMP Research
P.O. Box 8593
Knoxville, TN 37996-4800
(615) 573-1434 (Voice)
(615) 573-9197 (FAX)
Attn: Dr. Jeff Case

NAME
netstat

KEYWORDS
routing; IP; UNIX, VMS; free.

ABSTRACT
Netstat is a program that accesses network related data structures within the kernel, then provides an ASCII format at the terminal. Netstat can provide reports on the routing table, TCP connections, TCP and UDP ‘‘listens’’, and protocol memory management.

MECHANISM
Netstat accesses operating system memory to read the kernel routing tables.

CAVEATS
Kernel data structures can change while netstat is running.

BUGS
None known.

LIMITATIONS
None reported.

HARDWARE REQUIRED
No restrictions.

SOFTWARE REQUIRED
BSD UNIX or related OS, or VMS.

AVAILABILITY
Available via anonymous FTP from uunet.uu.net, in directory bsd-sources/src/ucb. Available with 4.xBSD UNIX and related operating systems. For VMS, available as part of TGV MultiNet IP software package, as well as Wollongong’s WIN/TCP.

NAME

netwatch

KEYWORDS

traffic; ethernet, IP, ring; eavesdrop; DOS; free.

ABSTRACT

PC/netwatch listens to an attached local broadcast network and displays one line of information for every packet that goes by. This information consists of the ‘‘to’’ and ‘‘from’’ local network addresses, the packet length, the value of the protocol type field, and 8 selected contiguous bytes of the packet contents. While netwatch is running it will respond to commands to display collected information, change its operating mode, or to filter for specific types of packets.

MECHANISM

Puts controller in promiscuous mode.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

The monitor can handle a burst rate of about 200 packets per second. Packets arriving faster than that are missed (but counted in the statistics of the network driver). The display rate is about 25 packets per second and there is a buffer that can hold 512 undisplayed packets. The monitor discards overflow packets.

HARDWARE REQUIRED

IBM PC compatible with CGA and network interface (3com 3C501, Interlan NI5010, or proNet p1300).

SOFTWARE REQUIRED

DOS 2.0 or higher, MicroSoft C (to generate custom executables)

AVAILABILITY

Available as a utility program in the pcip distribution from host husc6.harvard.edu, in directory pub/pcip. Available in a standalone package via anonymous FTP from windom.ucar.edu, in file pc/network/netwatch.arc; a binary ‘‘dearc’’ program is also available from windom.ucar.edu.

NAME

Network Integrator I

KEYWORDS

map, traffic; ethernet; UNIX.

ABSTRACT

This tool monitors traffic on network segments. All information is dumped to either a log file or, for real-time viewing, to a command tool window. Data is time-stamped according to date and time. Logging can continue for up to 24 hours.

The tool is flexible in data collection and presentation. Traffic filters can be specified according to header values of numerous protocols, including those used by Apple, DEC, Sun, HP, and Apollo. Bandwidth utilization can be monitored, as well as actual load and peak throughput. Additionally, the Network Integrator can analyze a network's topology, and record the location of all operational nodes on a network.

Data can be displayed in six separate formats of bar graphs. In addition, there are several routines for producing statistical summaries of the data collected.

MECHANISM

The tools work through RPC and XDR calls.

CAVEATS

Although the tool adds only little traffic to a network, generation of statistics from captured files requires a significant portion of a workstation's CPU.

BUGS

None known.

LIMITATIONS

Must be root to run monitor. There does not seem to be a limit to the number of nodes, since it monitors by segments. The only major limitation is the amount of disk space that a user can commit to the log files. The size of the log files, however, can be controlled through the tool's parameters.

HARDWARE REQUIRED

Sun3 or Sun4.

SOFTWARE REQUIRED

4.0BSD UNIX or greater, or related OS.

AVAILABILITY

Copyrighted, commercially available from
Network Integrators,
(408) 927-0412.

NAME

net_monitor

KEYWORDS

routing, status; DECnet, IP; curses, ping; UNIX, VMS; free, sourcelib.

ABSTRACT

Net_monitor uses ICMP echo (and DECnet reachability information on VAX/VMS) to monitor a network. The monitoring is very simplistic, but has proved useful. It periodically tests whether hosts are reachable and reports the results in a full-screen display. It groups hosts together in common sets. If all hosts in a set become unreachable, it makes a lot of racket with bells, since it assumes that this means that some common piece of hardware that supports that set has failed. The periodicity of the tests, hosts to test, and groupings of hosts are controlled with a single configuration file.

The idea for this program came from the PC/IP monitor facility, but is an entirely different program with different functionality.

MECHANISM

Reachability is tested using ICMP echo facilities for TCP/IP hosts (and DECnet reachability information on VAX/VMS). A DECnet node is considered reachable if it appears in the list of hosts in a ‘‘show network’’ command issued on a routing node.

CAVEATS

This facility has been found to be most useful when run in a window on a workstation rather than on a terminal connected to a host. It could be useful if ported to a PC (looks easy using FTP Software’s programming libraries), but this has not been done. Curses is very slow and cpu intensive on VMS, but the tool has been run in a window on a VAXstation 2000. Just don’t try to run it on a terminal connected to a 11/750.

BUGS

None known.

LIMITATIONS

This tool is not meant to be a replacement for a more comprehensive network management facility such as is provided with SNMP.

HARDWARE REQUIRED

A host with a network connection.

SOFTWARE REQUIRED

Curses, 4.xBSD UNIX socket programming libraries (limited set) and some flavor of TCP/IP that supports ICMP echo request (ping). It has been run on VAX/VMS running WIN/TCP and several flavors of 4BSD UNIX (including SunOS 3.2, 4.0, and 4.3BSD). It could be ported to any platform that provides a BSD-style programming library with an ICMP echo request facility and curses.

AVAILABILITY

Requests should be sent to the author:

Dale Smith
Asst Dir of Network Services
University of Oregon
Computing Center
Eugene, OR 97403-1211

Internet: dsmith@oregon.uoregon.edu.
BITNET: dsmith@oregon.bitnet
UUCP: ...hp-pcd!uoregon!dsmith
Voice: (503)686-4394

With the source code, a makefile is provided for most any UNIX box and a VMS makefile compatible with the make distributed with PMDF. A VMS DCL command file is also provided, for use by those VMS sites without ‘‘make.’’

The author will attempt to fix bugs, but no support is promised. The tool is copyrighted, but free (for now).

NAME
nfswatch

KEYWORDS
traffic; ethernet, IP, NFS; curses, eavesdrop; UNIX; free.

ABSTRACT
Nfswatch monitors all incoming ethernet traffic to an NFS file server and divides it into several categories. The number and percentage of packets received in each category is displayed on the screen in a continuously updated display.

All exported file systems are monitored by default. Other files may optionally be monitored. Options also allow monitoring of traffic destined for a remote host instead of the local host, or monitoring traffic sent by a single host. Items such as the sample interval length can be adjusted either on the command line or interactively. Facilities for taking screen ‘‘snapshots,’’ saving all data to a log file, and summarizing the log file are included. Nfslogsum, a program that summarizes the log file, is included in the distribution.

MECHANISM
Nfswatch uses the Network Interface Tap in promiscuous mode to monitor the ethernet. It filters out NFS packets destined for the local (or remote) host, and then decodes the file handles in order to determine which file or file system a request pertains to.

CAVEATS
Because the NFS file handle is a non-standard (server private) piece of data, the file system monitoring part of the program will break whenever the format of a file handle is not what it expects to see. This is easily fixed in the code, however. The code presently understands SunOS 4.0 file handles.

BUGS
None known.

LIMITATIONS
Up to 256 exported file systems and 256 individual files can be monitored, but only (2 * (DisplayLines - 16)) will be displayed on the screen (all data will be written to the log file).

Only NFS requests made by client machines are counted; the NFS traffic generated by the server in response to these requests is not counted.

HARDWARE REQUIRED
Has been tested on Sun-3 and Sun-4 systems. No hardware dependencies, but see below.

SOFTWARE REQUIRED
SunOS 4.0 or higher. The STREAMS NIT device is used. Fairly easy code modifications should be able to make it run under older SunOS releases, or other versions of BSD UNIX with a NIT-like device.

AVAILABILITY
Copyrighted, but freely distributable. Available via anonymous FTP from hosts icarus.riacs.edu and spam.itstd.sri.com in pub/nfswatch.tar.Z. There should also be a copy on the 1989 Sun User's Group tape.

NAME
nhfsstone

KEYWORDS
benchmark, generator; NFS; spoof; UNIX; free.

ABSTRACT
Nhfsstone (pronounced n-f-s-stone, the ‘‘h’’ is silent) is an NFS benchmarking program. It is used on an NFS client to generate an artificial load with a particular mix of NFS operations. It reports the average response time of the server in milliseconds per call and the load in calls per second. The nhfsstone distribution includes a script, ‘‘nhfsnums’’ that converts test results into plot(5) format so that they can be graphed using graph(1) and other tools.

MECHANISM
Nhfsstone is an NFS traffic generator. It adjusts its calling patterns based on the client’s kernel NFS statistics and the elapsed time. Load can be generated over a given time or number of NFS calls.

CAVEATS
Nhfsstone will compete for system resources with other applications.

BUGS
None known.

LIMITATIONS
None reported.

HARDWARE REQUIRED
No restrictions.

SOFTWARE REQUIRED
4.xBSD-based UNIX

AVAILABILITY
Available via anonymous FTP from bugs.cs.wisc.edu. Alternatively, Legato Systems will provide the program free of charge, if certain conditions are met. Send name and both email and U.S. mail addresses to:

Legato Systems, Inc.
Nhfsstone
260 Sheridan Avenue
Palo Alto, California 94306

A mailing list is maintained for regular information and bug fixes: nhfsstone@legato.com or uunet!legato.com!nhfsstone. To join the list: nhfsstone-request@legato.com or uunet!legato.com!nhfsstone-request.

NAME

NMC — the Hughes LAN Systems 9100 Network Management Center

KEYWORDS

control, manager, routing, status, traffic; bridge, DECnet, ethernet, IP; NMS, SNMP; UNIX.

ABSTRACT

The 9100 Network Management Center provides the capability to manage and control standards-based networking products from Hughes LAN Systems' and other vendors. This management extends to all network products that are equipped with the industry standard SNMP (Simple Network Management Protocol). A comprehensive relational database manages the data and ensures easy access and control of resources throughout the network.

9100 NMC software provides the following functions:

Database Management

Stores and retrieves the information required to administer and configure the network. It can be used to:

Store and recall configuration data for all devices.
Provide availability history for devices.
Provides full-function SQL interface.
Assign new internet addresses.
Provide administrative information such as physical location of devices, person responsible, maintenance history, asset data, hardware/software versions, etc.

Configuration Management

A comprehensive configuration model that enables you to:

Retrieve configuration information from SNMP devices.
Configure HLS devices using SNMP.
Configures attributes relating to TCP/IP, DECnet and other protocols in HLS devices using SNMP.
Poll devices to compare their current attribute values with those in the database and produce reports of the discrepancies.
Collect data about the state of the network.

Performance Management

Displays local network traffic graphically, by packet size, protocol, network utilization, sources and destinations of packets, etc.

Fault Management

Provides availability monitoring and indicates potential problems.

Scheduled availability monitoring of devices.
SNMP traps (alarms) are recorded in an alarm log.
New alarms are indicated by a flashing icon and optional audio alert.
Possible causes and suggested actions for the alarms are listed.
Cumulative reports can be produced.

Utilities Function

Allows you to view and/or stop existing NMC processes, and to define schedules for invoking NMC applications and database maintenance utilities.

MECHANISM

SNMP.

CAVEATS

None reported.

BUGS

None known.

LIMITATIONS

Maximum number of nodes that can be monitored is 18,000. This can include Hosts, Terminal Servers, PCs, and Bridges.

HARDWARE REQUIRED

The host for the NMC software is a Sun 3 desktop workstation. Recommended minimum hardware is the Sun 3/80 Color with a 1/4" SCSI tape drive.

SOFTWARE REQUIRED

The NMC, which is provided on 1/4" tape format, runs on the Sun 4.0 Operating System.

AVAILABILITY

A commercial product of:

Hughes LAN Systems Inc.
1225 Charleston Road
Mountain View, CA 94043
Phone: (415) 966-7300
Fax: (415) 960-3738
RCA Telex: 276572

NAME

NNStat

KEYWORDS

manager, status, traffic; ethernet, IP; eavesdrop, NMS; UNIX; free.

ABSTRACT

NNStat is a collection of programs that provides an internet statistic collecting capability. The NNStat strategy for statistic collection is to collect traffic statistics via a promiscuous ethernet tap on the local networks, versus instrumenting the gateways. If all traffic entering or leaving a network or set of networks traverses a local ethernet, then by stationing a statistic gathering agent on each local network a profile of network traffic can be gathered. Statistical data is retrieved from the local agents by a global manager.

A program called ''statspy'' performs the data gathering function. Essentially, statspy reads all packets on an ethernet interface and records all information of interest. Information of interest is gathered by examining each packet and determining if the source or destination IP address is one that is being monitored, typically a gateway address. If so then the contents of the packet are examined to see if they match further criteria.

A program called ''collect'' performs global data collection. It periodically polls various statspy processes in the domain of interest to retrieve locally logged statistical data.

The NNSTAT distribution comes with several sample awk programs which process the logged output of the collect program.

MECHANISM

Local agents (statspy processes) collect raw traffic data via a promiscuous ethernet tap. Statistical, filtered or otherwise reduced data is retrieved from the local agents by a global manager (the ''collect'' process).

CAVEATS

None.

BUGS

Bug fixes, extensions, and other pointers are discussed in the electronic mail forum, bytecounters. To join, send a request to bytecounters-request@venera.isi.edu. Forum exchanges are archived in the file bytecounters/bytecounters.mail, available via anonymous FTP from venera.isi.edu.

LIMITATIONS

NNStat presumes a topology of one or more long haul networks gatewayed to local ethernets.

A kernel mod required to run with SunOS4. These mods are described in the bytecounters archive.

HARDWARE REQUIRED

Ethernet interface. Sun 3, Sun 4 (SPARC), or PC RT workstation.

SOFTWARE REQUIRED

Distribution is for BSD UNIX, could easily be adapted to any UNIX with promiscuous ethernet support.

AVAILABILITY

Distribution is available via anonymous FTP from venera.isi.edu, in file pub/NNStat.tar.Z. Documentation is in pub/NNStat.userdoc.ms.Z.

NAME

NPRV — IP Node/Protocol Reachability Verifier

KEYWORDS

map, routing, status; IP; ping; VMS; free.

ABSTRACT

NPRV is a full-screen, keypad-oriented utility that runs under VAX/VMS. It allows the user to quickly scan through a user-defined list of IP addresses (or domain names) and verify a node's reachability. The node's reachability is determined by performing an ICMP echo, UDP echo and a TCP echo at alternating three second intervals. The total number of packets sent and received are displayed, as well as the minimum, average and maximum round-trip times (in milliseconds) for each type of echo. Additionally, a ''trace route'' function is performed to determine the path from the local system to the remote host. Once all of the trace route information has filled the screen, a ''snapshot'' of the screen can be written to a text file. Upon exiting the utility, these text files can be used to generate a logical network map showing host and gateway interconnectivity.

MECHANISM

The ICMP echo is performed by sending ICMP ECHO REQUEST packets. The UDP and TCP echoes are performed by connecting to the UDP/TCP echo ports (port number 7). The trace route information is compiled by sending alternating ICMP ECHO REQUEST packets and UDP packets with very large destination UDP port numbers (in two passes). Each packet is initially sent with a TTL (time to live) of 1. This should cause an ICMP TIME EXCEEDED error to be generated by the first routing gateway. Then each packet is sent with a TTL of 2. This should cause an ICMP TIME EXCEEDED error to be generated by the second routing gateway. Then each packet is sent with a TTL of 3, and so on. This process continues until an ICMP ECHO REPLY or UDP PORT UNREACHABLE is received. This indicates that the remote host has been reached and that the trace route information is complete.

CAVEATS

This utility sends one echo packet per second (ICMP, UDP or TCP), as well as sending out one trace route packet per second. If a transmitted trace route packet is returned in less than one second, another trace route packet is sent in 100 milliseconds. This could cause a significant amount of contention on the local network.

BUGS

None known. Please report any discovered bugs to the author at:

Allen Sturtevant
National Magnetic Fusion Energy Computer Center
Lawrence Livermore National Laboratory
P.O. Box 808; L-561
Livermore, CA 94550
Phone : (415) 422-8266
E-Mail: sturtevant@ccc.nmfecc.gov

LIMITATIONS

The user is required to have SYSPRV privilege to perform the ICMP Echo and trace route functions. The utility will still run with this privilege disabled, but only the UDP Echo and TCP Echo information will be displayed. This utility is written in C, but unfortunately it cannot be easily ported over to UNIX since many VMS system calls are used and all screen I/O is done using the VMS Screen Management Routines.

HARDWARE REQUIRED

Any network interface supported by TGV Incorporated's MultiNet software.

SOFTWARE REQUIRED

VAX/VMS V5.1+ and TGV Incorporated's MultiNet version 2.0.

AVAILABILITY

For executables only, FTP to the ANONYMOUS account (password GUEST) on CCC.NMFECC.GOV (128.55.128.30) and GET the following files:

[ANONYMOUS.PROGRAMS.NPRV]NPRV.DOC	(ASCII text)
[ANONYMOUS.PROGRAMS.NPRV]NPRV.EXE	(binary)
[ANONYMOUS.PROGRAMS.NPRV]SAMPLE.IPA	(ASCII text)

NAME

nslookup

KEYWORDS

status; DNS; spoof; UNIX, VMS; free.

ABSTRACT

Nslookup is a program used for interactive query of ARPA Internet domain servers. This program is useful for diagnosing routing or mail delivery problems, where often a local domain server is responding with an incorrect internet address. It is essentially a database front end which converts user queries into domain name queries. By default nslookup queries the local domain name server but you can specify additional servers. Additional information beyond the mapping of domain names to internet addresses is possible.

MECHANISM

Formats and sends domain name queries.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None known.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

BSD UNIX or related OS, or VMS.

AVAILABILITY

Nslookup is part of the ‘‘named’’ distribution, available via anonymous FTP from uunet.uu.net, in directories bsd-sources/src/etc and bsd-sources/src/network, and part of the ‘‘bind’’ distribution, available via anonymous FTP from ucbarpa.berkeley.edu, in directory 4.3. Also available with 4.xBSD UNIX and related operating systems. For VMS, available as part of TGV MultiNet IP software package, as well as Wollongong’s WIN/TCP.

NAME

osilog — OSI event Logger

KEYWORDS

alarm, manager; OSI; UNIX; free.

ABSTRACT

The osilog program receives management event reports for the operation of the ISODE Transport layer (ISO Transport Protocol class 0) on one or more managed systems, formats them suitably to facilitate post-processing and records them for future analysis.

MECHANISM

It communicates with the System Management Agents (SMAs) on the selected systems via CMIP.

CAVEATS

The System Management Agent (SMA) must be running on the hosts selected to provide management reports.

BUGS

None known.

LIMITATIONS

ISODE Transport Layer only supported by the SMA at present.

HARDWARE REQUIRED

Developed and tested on Sun3.

SOFTWARE REQUIRED

The ISODE protocol suite, BSD UNIX.

AVAILABILITY

The osilog and related tools, known as OSIMIS (OSI Management Information Service), are publicly available from University College London, England via FTP and FTAM. To obtain information regarding a copy send email to gknight@ac.ucl.cs.uk or call +44 1 380 7366.

NAME

osimic — OSI Microscope

KEYWORDS

manager, status; OSI; UNIX; free.

ABSTRACT

The osimic program is a human user interface to the management information base on the ISODE Transport layer (ISO Transport Protocol class 0). It allows browsing through the management information tree and enables the manipulation of attribute values. It is implemented using the SunView package of the SunTools window system.

MECHANISM

It communicates with the System Management Agent (SMA) on the selected system via CMIP.

CAVEATS

The System Management Agent (SMA) must be running on the host where the mib is being examined.

BUGS

None known.

LIMITATIONS

ISODE Transport Layer only supported by the SMA at present.

HARDWARE REQUIRED

Developed and tested on Sun3.

SOFTWARE REQUIRED

The ISODE protocol suite, BSD UNIX, SunView/SunTools.

AVAILABILITY

The osimic and related tools, known as OSIMIS (OSI Management Information Service), are publicly available from University College London, England via FTP and FTAM. To obtain information regarding a copy send email to gknight@ac.ucl.cs.uk or call +44 1 380 7366.

NAME
osimon — OSI Monitor

KEYWORDS
manager, status, traffic; OSI; curses; UNIX; free.

ABSTRACT
The osimon program monitors activity of the ISODE Transport layer (ISO Transport Protocol class 0), displaying entries for the active transport entities and connections. The display is dynamically updated in the case of significant events such as connection opening and closing and packet traffic, as information is received in the form of event reports from a SMA. It uses the UNIX curses package for screen management.

MECHANISM
It communicates with the System Management Agent (SMA) on the selected system via CMIP.

CAVEATS
The System Management Agent (SMA) must be running on the host being monitored.

BUGS
For the terminal type Sun, there are some transient problems with the display.

LIMITATIONS
ISODE Transport Layer only supported at present.

HARDWARE REQUIRED
Developed and tested on Sun3 for various terminal types.

SOFTWARE REQUIRED
The ISODE protocol suite, BSD UNIX.

AVAILABILITY
The osimon and related tools, known as OSIMIS (OSI Management Information Service), are publicly available from University College London, England via FTP and FTAM. To obtain information regarding a copy send email to gknight@ac.ucl.cs.uk or call +44 1 380 7366.

NAME

OSITRACE

KEYWORDS

traffic; OSI; eavesdrop; UNIX; free.

ABSTRACT

OSITRACE is a network performance tool that displays information about ISO TP4 connections. One line of output is displayed for each packet indicating the time, source, destination, length, packet type, sequence number, credit, and any optional parameters contained in the packet. Numerous options are available to control the output of OSITRACE.

To obtain packets to analyze, OSITRACE uses Sun Microsystems' Network Interface Tap (NIT) in SunOS 3.4, 3.5, and 4.0.X. OSITRACE may also obtain data from the NETMON utility which is described as another tool entry.

In Sun systems, OSITRACE may be easily installed: OSI kernel support is not needed, nor is any other form of OSI software support.

MECHANISM

This tool has been designed in such a way that code to process different protocol suites may be easily added. As such, OSITRACE also has the ability to trace the DOD TCP protocols.

CAVEATS

None.

BUGS

Bug reports and questions should be addressed to: ie-tools@gateway.mitre.org

Requests to join this mailing list: ie-tools-request@gateway.mitre.org

Questions and suggestions can also be directed to: Greg Hollingsworth, gregh@gateway.mitre.org

LIMITATIONS

None reported.

HARDWARE REQUIRED

No restriction.

SOFTWARE REQUIRED

SunOS 3.4, 3.5, or 4.0.X, or BSD UNIX-like network protocols with NETMON installed.

AVAILABILITY

OSITRACE is copyrighted by the MITRE-Washington Networking Center, but freely distributed ''as is.'' It requires retention of a copyright text in code derived from it. The distribution is available by anonymous FTP in pub/pdutrace.tar or pub/pdutrace.tar.Z from aelred-3.ie.org.

NAME

OverVIEW

KEYWORDS

manager, status; IP; NMS, SNMP; DOS.

ABSTRACT

Network and internet monitor; Performance monitor; Fully Graphic user interface; Event logging; TFTP boot server

MECHANISM

OverVIEW uses SNMP to query routers, gateways and hosts. Also supports SGMP, PING and is committed to CMIP/CMOT. The SNMP queries allow dynamic determination of configuration and state. Sets of related queries allows monitoring of congestion and faults. The hardware and software are sold as an integrated package.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

256 nodes, 256 nets

HARDWARE REQUIRED

80286, 640K, EGA, mouse.

SOFTWARE REQUIRED

MS-DOS, OverVIEW, Network kernel, Mouse driver, SNMP agents for monitored devices.

AVAILABILITY

Fully supported product of Proteon, Inc. For more information, contact:

Proteon, Inc.	Phone: (508) 898-2800
2 Technology Drive	Fax: (508) 366-8901
Westborough, MA 01581	Telex: 928124

NAME
ping

KEYWORDS
generator, status; IP; ping; DOS, UNIX, VMS; free.

ABSTRACT
Ping is perhaps the most basic tool for internet management. It verifies that a remote IP implementation and the intervening networks and interfaces are functional. It can be used to measure round trip delay. Numerous versions of the ping program exist.

MECHANISM
Ping is based on the ICMP ECHO_REQUEST message.

CAVEATS
If run repeatedly, ping could generate high system loads.

BUGS
None known.

LIMITATIONS
PC/TCP's ping is the only implementation known support both loose and strict source routing. Though some ping implementations support the ICMP ''record route'' feature, the usefulness of this option for debugging routes is limited by the fact that many gateways do not correctly implement it.

HARDWARE REQUIRED
No restrictions.

SOFTWARE REQUIRED
None.

AVAILABILITY
Ping is widely included in TCP/IP distributions. Public domain versions of ping are available via anonymous FTP from uunet.uu.net, in directory bsd-sources/src/etc, and from venera.isi.edu, in directory pub.

NAME

proxyd — SNMP proxy agent daemons from SNMP Research.

KEYWORDS

control, status; bridge, ethernet, IP, ring, star; NMS, SNMP; UNIX; library, sourcelib.

ABSTRACT

SNMP proxy agents may be used to permit the monitoring and controlling of network elements which are otherwise not addressable using the SNMP management protocol (e.g., a network bridge that implements a proprietary management protocol). Similarly, SNMP proxy agents may be used to protect SNMP agents from redundant network management agents through the use of caches. Finally, SNMP proxy agents may be used to implement elaborate MIB access policies. The proxy agent daemon listens for SNMP queries and commands from logically remote network management stations, translates and retransmits those as appropriate network management queries or cache lookups, listens for and parses the responses, translates the responses into SNMP responses, and returns those responses as SNMP messages to the network management station that originated the transaction. The proxy agent daemon also emits SNMP traps to identified trap receivers. The proxy agent daemon is architected to make the addition of additional vendor-specific variables a straight-forward task. The proxy application comes complete with source code including a powerful set of portable libraries for generating and parsing SNMP messages and a set of command line utilities.

MECHANISM

Network management variables are made available for inspection and/or alteration by means of the Simple Network Management Protocol (SNMP).

CAVEATS

None.

BUGS

None known.

LIMITATIONS

This application is a template for proxy application writers.

Only a few of the many LanBridge 100 variables are supported.

HARDWARE REQUIRED

System from Sun Microsystems, Incorporated.

SOFTWARE REQUIRED

Sun OS 3.5 or 4.x

AVAILABILITY

This is a commercial product available under license from:

SNMP Research
P.O. Box 8593
Knoxville, TN 37996-4800
(615) 573-1434 (Voice)
(615) 573-9197 (FAX)
Attn: Dr. Jeff Case

NAME

query, ripquery

KEYWORDS

routing; IP; spoof; UNIX; free.

ABSTRACT

Query allows remote viewing of a gateway's routing tables.

MECHANISM

Query formats and sends a RIP request or POLL command to a destination gateway.

CAVEATS

Query is intended to be used a a tool for debugging gateways, not for network management. SNMP is the preferred protocol for network management.

BUGS

None known.

LIMITATIONS

The polled gateway must run RIP.

HARDWARE REQUIRED

No restriction.

SOFTWARE REQUIRED

4.3BSD UNIX or related OS.

AVAILABILITY

Available with routed and gated distributions.

Routed may be obtained via anonymous FTP from uunet.uu.net, in file bsd-sources/src/network/routed.tar.Z.

Gated may be obtained via anonymous FTP from devvax.tn.cornell.edu. Distribution files are in directory pub/gated.

NAME

SERAG — the Simple Event Reporting and Alarm Generation tool

KEYWORDS

alarm, security; ethernet, IP; NMS, proprietary; UNIX.

ABSTRACT

The Simple Event Reporting and Alarm Generation (SERAG) collects error messages and other event reports from servers on a LAN. Any node with UDP/IP can be the source of such messages/reports. The logging of error messages is integrated with the audit trail facility of the Network Control Server (NCS) from 3COM. Alarms are generated on the NCS based on predefined conditions. Alarms may be sent to the console of the NCS, logged in a file, or routed via WAN to a service center.

SERAG can automatically detect a predefined set of errors in the servers and generate alarms. The breakdown of a server in the LAN may also result in alarm generation.

SERAG creates an error log that can be used for post-testing analysis.

MECHANISM

The tool searches through the audit trail (error log) files for events specified by the user. The search may be constrained to specific nodes in the network and to a specific time frame. Events may be combined into conditions which are logical expressions (e.g., look for eventA and eventB and not eventC within time frame so and so). This is an interactive query facility to analyze the audit trail (error log).

The user may also ask for such conditions to be checked at regular intervals, and specify routing of error messages in case the condition is satisfied. The checking of such conditions is done by a daemon process running in the background.

CAVEATS

May impact the performance of the NCS if error logs are big, or if conditions are computationally complex.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

A workstation running UNIX.

SOFTWARE REQUIRED

Implemented in C (using lex and yacc) on a Sun 3/50. Also runs under Xenix. Should work with most versions of UNIX.

AVAILABILITY

Developed jointly by ELAB-RUNIT and Norsk Data:

Tor Didriksen, Ole-Hjalmar Kristensen, Steinar Haug,
Eldfrid Oefsti Oevstedal, Tor Staalhane
ELAB-RUNIT
N-7034 Trondheim
Norway

phone: +47 7 593000
fax : +47 7 532586
email: didrik@idt.unit.no
sthaug@idt.unit.no
kristensen@vax.runit.unit.no

Commercially available from:

Norsk Data A/S
P.O. Box 25, Bogerud
N-0621 Oslo 6
Norway
ref: network management/security management/fault management

phone: +47 2 627500
fax : +47 2 296796

NAME
sma — OSI System Management Agent

KEYWORDS
alarm, manager, status; OSI; UNIX; free.

ABSTRACT
The sma is a CMIP agent which runs on BSD UNIX and provides access to management information on the operation of the ISODE transport layer (ISO Transport Protocol class 0). It also supports the sending of event reports. Activity can be recorded in a log file.

MECHANISM
The sma communicates with the active ISODE transport entities using UNIX UDP sockets in order to receive the management information which is made available to other manager processes via CMIP.

CAVEATS
None.

BUGS
None known.

LIMITATIONS
ISODE Transport Layer only supported at present.

HARDWARE REQUIRED
Developed on Sun3, tested on Sun3 and VAXStation.

SOFTWARE REQUIRED
The ISODE protocol suite, BSD UNIX.

AVAILABILITY
The sma and related tools, known as OSIMIS (OSI Management Information Service), are publicly available from University College London, England via FTP and FTAM. To obtain information regarding a copy send email to gknight@ac.ucl.cs.uk or call +44 1 380 7366.

NAME

Sniffer

KEYWORDS

analyzer, generator, traffic; DECnet, ethernet, IP, NFS, OSI, ring, SMTP, star; eavesdrop; standalone.

ABSTRACT

The Network General Sniffer is a protocol analyzer for performing LAN diagnostics, monitoring, traffic generation, and troubleshooting. The Sniffer protocol analyzer has the capability of capturing every packet on a network and of decoding all seven layers of the OSI protocol model. Capture frame selection is based on several different filters: protocol content at lower levels; node addresses; pattern matching (up to 8 logically-related patterns of 32 bytes each); and destination class. Users may extend the protocol interpretation capability of the Sniffer by writing their own customized protocol interpreters and linking them to the Sniffer software.

The Sniffer displays network traffic information and performance statistics in real time, in user-selectable formats. Numeric station addresses are translated to symbolic names or manufacturer ID names. Network activities measured include frames accepted, Kbytes accepted, and buffer use. Each network version has additional counters for activities specific to that network. Network activity is expressed as frames/second, Kbytes/second, or per cent of network bandwidth utilization.

Data collection by the Sniffer may be output to printer or stored to disk in either print-file or spread-sheet format.

Protocol suites understood by the Sniffer include: Banyan Vines, IBM Token-Ring, Novell Netware, XNS/MS-Net (3Com 3+), DECnet, TCP/IP (including SNMP and applications-layer protocols such as FTP, SMTP, and TELNET), X Windows (for X version 11), NFS, and several SUN proprietary protocols (including mount, pmap, RPC, and YP). Supported LANs include: ethernet, Token-ring (4Mb and 16Mb versions), ARCNET, StarLAN, IBM PC Network (Broadband), and Apple Localtalk Network.

MECHANISM

The Sniffer is a self-contained, portable protocol analyzer that require only AC line power and connection to a network to operate. Normally passive (except when in Traffic Generator mode), it captures images of all or of selected frames in a working buffer, ready for immediate analysis and display.

The Sniffer is a standalone device. Two platforms are available: one for use with single network topologies, the other for use with multi-network topologies. Both include Sniffer core software, a modified network interface card (or multiple cards), and optional protocol interpreter suites.

All Sniffer functions may be remotely controlled from a modem-connected PC. Output from the Sniffer can be imported to database or spreadsheet packages.

CAVEATS

In normal use, the Sniffer is a passive device, and so will not adversely effect network performance. Performance degradation will be observed, of course, if the Sniffer is set to Traffic Generator mode and connected to an active network.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

None. The Sniffer is a self-contained unit, and includes its own interface card. It installs into a network as would any normal workstation.

SOFTWARE REQUIRED

None.

AVAILABILITY

The Sniffer is available commercially. For information on your local representative, call or write:

> Network General Corporation 4200 Bohannon Drive Menlo Park, CA 94025 Phone: (415) 688-2700
> Fax: 415-321-0855

For acquisition by government agencies, the Sniffer is included on the GSA schedule.

NAME

The SNMP Development Kit

KEYWORDS

manager, status; IP; NMS, SNMP; UNIX; free, sourcelib.

ABSTRACT

The SNMP Development Kit comprises C Language source code for a programming library that facilitates access to the management services of the SNMP (RFC 1098). Sources are also included for a few simple client applications whose main purpose is to illustrate the use of the library. Example client applications query remote SNMP agents in a variety of modes, and generate or collect SNMP traps. Code for an example SNMP agent that supports a subset of the Internet MIB (RFC 1066) is also included.

MECHANISM

The Development Kit facilitates development of SNMP-based management applications — both clients and agents. Example applications execute SNMP management operations according to the values of command line arguments.

CAVEATS

None.

BUGS

Fixed in the next release.

LIMITATIONS

None reported.

HARDWARE REQUIRED

The SNMP library source code is highly portable and runs on a wide range of platforms.

SOFTWARE REQUIRED

The SNMP library source code has almost no operating system dependencies and runs in a wide range of environments. Certain portions of the example SNMP agent code are specific to the 4.3BSD implementation of the UNIX system for the DEC MicroVAX.

AVAILABILITY

The Development Kit is available via anonymous FTP from host allspice.lcs.mit.edu. The copyright for the Development Kit is held by the Massachusetts Institute of Technology, and the Kit is distributed without charge according to the terms set forth in its code and documentation. The distribution takes the form of a UNIX tar file.

Bug reports, questions, suggestions, or complaints may be mailed electronically to snmp-dk@ptt.lcs.mit.edu, although no response in any form is guaranteed. Distribution via UUCP mail may be arranged by contacting the same address. Requests for hard-copy documentation or copies of the distribution on magnetic media are never honored.

NAME

Snmp Libraries and Utilities from SNMP Research.

KEYWORDS

alarm, control, manager, map, routing, status; bridge, DECnet, ethernet, IP, OSI, ring, star; NMS, SNMP; DOS, UNIX, VMS; sourcelib.

ABSTRACT

The SNMP Libraries and Utilities serve two purposes:

1) to act as building blocks for the construction of SNMP-based agent and manager applications; and

2) to act as network management tools for network fire fighting and report generation.

The libraries perform ASN.1 parsing and generation tasks for both network management station applications and network management agent applications. These libraries hide the details of ASN.1 parsing and generation from application writers and make it unnecessary for them to be expert in these areas. The libraries are very robust with considerable error checking designed in. The several command line utilities include applications for retrieving one or many variables, retrieving tables, or effecting commands via the setting of remote network management variables.

MECHANISM

The parsing is performed via recursive descent methods. Messages are passed via the Simple Network Management Protocol (SNMP).

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

This software has been ported to a wide range of systems, too numerous to itemize. It includes workstations, general purpose timesharing systems, and embedded hardware in intelligent network devices such as repeaters, bridges, and routers.

SOFTWARE REQUIRED

C compiler, TCP/IP library from a variety of sources.

AVAILABILITY

This is a commercial product available under license from:

SNMP Research
P.O. Box 8593
Knoxville, TN 37996-4800
(615) 573-1434 (Voice)
(615) 573-9197 (FAX)
Attn: Dr. Jeff Case

NAME
: snmpask

KEYWORDS
: manager, status; IP; NMS, SNMP; UNIX.

ABSTRACT
: Snmpask is a network monitoring application which gathers specific information from a single network entity at regular intervals and stores this information into UNIX flat files. A report generation package is included in the NYSERNet SNMP Software Distribution to produce reports and graphs from the raw data.

MECHANISM
: Snmpask uses SNMP to gather its information. The agent which must be queried and the variables to query for are specified in a configuration file.

CAVEATS
: An SNMP agent must be running in the network entity being monitored in order for snmpask to be useful.

BUGS
: None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
: SNMP polling is done synchronously. Only a single agent can be polled per snmpask process. Only 16 variables can be requested per snmpask process.

HARDWARE REQUIRED
: Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
: Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY
: Snmpask is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmpd

KEYWORDS

manager, status; IP; NMS, SNMP; UNIX.

ABSTRACT

Snmpd is an SNMP agent which runs on UNIX derivatives and answers network management queries from network management stations supporting SNMP. Snmpd also supports the sending of SNMP traps.

MECHANISM

Snmpd conforms to SNMP as specified in RFC 1098. Certain user configurable options are manipulated through a simple configuration file.

CAVEATS

UNIX does not support all of the MIB variables specified in RFC 1066. Snmpd does the best it can to find the answers.

BUGS

None outstanding. They are fixed as reports come in. report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS

See CAVEATS.

HARDWARE REQUIRED

Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED

Some UNIX variant.

AVAILABILITY

Snmpd is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmpd — an SNMP host/gateway agent daemon from SNMP Research.

KEYWORDS

manager, status; bridge, ethernet, IP, ring, star; NMS, SNMP; DOS, UNIX; sourcelib.

ABSTRACT

The snmpd agent daemon listens for and responds to network management queries and commands from logically remote network management stations. The agent daemon also emits SNMP traps to identified trap receivers. The agent daemon is architected to make the addition of additional vendor-specific variables a straight-forward task. The snmpd application comes complete with source code including a powerful set of portable libraries for generating and parsing SNMP messages and a set of command line utilities.

MECHANISM

Network management variables are made available for inspection and/or alteration by means of the Simple Network Management Protocol (SNMP).

CAVEATS

None.

BUGS

None known.

LIMITATIONS

Only the operating system variables available without source code modifications to the operating system and device device drivers are supported.

HARDWARE REQUIRED

This software has been ported to a wide range of systems, too numerous to itemize. It includes workstations, general purpose timesharing systems, and embedded hardware in intelligent network devices such as repeaters, bridges, and routers.

SOFTWARE REQUIRED

C compiler, ‘‘.h’’ files for operating system.

AVAILABILITY

This is a commercial product available under license from:

SNMP Research
P.O. Box 8593
Knoxville, TN 37996-4800
(615) 573-1434 (Voice)
(615) 573-9197 (FAX)
Attn: Dr. Jeff Case

NAME
snmplookup

KEYWORDS
manager, status; IP; NMS, SNMP; UNIX.

ABSTRACT
Snmplookup is a network monitoring application that allows the interactive querying of a network entity. Snmplookup mimics nslookup, the DNS interactive query tool, in style and feel.

MECHANISM
Snmplookup uses SNMP to gather its information. The network entity to be queried and the variable to be retrieved can be entered from the command shell after snmplookup is invoked.

CAVEATS
An SNMP agent must be running on the network entity being monitored.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
See CAVEATS.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY
Snmplookup is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmpperfmon

KEYWORDS

manager, status; IP; curses, NMS, SNMP; UNIX.

ABSTRACT

Snmpperfmon is a network monitoring application based on the Berkeley curses terminal graphics package and the Simple Network Management Protocol. The application monitors certain interface statistics from a single agent and displays them in tabular form on a standard terminal screen.

MECHANISM

Snmpperfmon uses SNMP to gather its information. The agent to be queried is specified on the command line.

CAVEATS

An SNMP agent must be running in the network entity being monitored in order for snmpperfmon to be useful.

BUGS

None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS

SNMP polling is done synchronously. Only the predetermined (read ‘‘hard coded’’) interface statistics can be displayed.

HARDWARE REQUIRED

Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED

Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library. The ‘‘curses’’ library.

AVAILABILITY

Snmpperfmon is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmppoll

KEYWORDS

manager, status; IP; NMS, SNMP; UNIX.

ABSTRACT

Snmppoll is a network monitoring application which gathers specific information from a network at regular intervals and stores this information into UNIX flat files. A report generation package is included in the NYSERNet SNMP Software Distribution to produce reports and graphs of raw data collected via SNMP.

MECHANISM

Snmppoll uses SNMP to gather its information. The agents which must be queried and the variables to query for are specified in a configuration file.

CAVEATS

An SNMP agent must be running in the network entity being monitored in order for snmppoll to be useful.

BUGS

None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS

SNMP polling is done synchronously.

HARDWARE REQUIRED

Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED

Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY

Snmppoll is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmpquery

KEYWORDS

manager, status; IP; NMS, SNMP; UNIX.

ABSTRACT

Snmpquery is a network monitoring application which allows the simple query of a single network entity from the command line.

MECHANISM

Snmpquery uses SNMP to gather its information. The entity to be monitored and the variables to be retrieved must be specified on the command line.

CAVEATS

An SNMP agent must be running on the network entity being monitored.

BUGS

None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS

Only one network entity can be managed per invocation.

HARDWARE REQUIRED

Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED

Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY

Snmpquery is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmproute

KEYWORDS
manager, routing; IP; NMS, SNMP; UNIX.

ABSTRACT
Snmproute is a network monitoring application that allows the user to query for the entire routing table or a single routing table entry from a network entity.

MECHANISM
Snmproute uses SNMP to gather its information. The network entity to be queried and the destination network to be queried for must be specified on the command line.

CAVEATS
An SNMP agent must be running on the network entity being monitored.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
Only one network entity can be queried per invocation.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY
Snmproute is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmpset

KEYWORDS

control, manager; IP; NMS, SNMP; UNIX.

ABSTRACT

Snmpset is a network management application that allows the alteration of a single variable in a specific agent.

MECHANISM

Snmpset uses SNMP to alter the agent variables. The agent to which the set is directed and the variable to alter must be specified on the command line. The user is prompted before any changes are made.

CAVEATS

An SNMP agent must be running in the network entity being managed in order for snmpset to be useful. In addition, a read-write community must be configured on the agent.

BUGS

None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS

Only one variable can be altered per invocation.

HARDWARE REQUIRED

Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED

Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY

Snmpset is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmpsrc

KEYWORDS
manager, routing; IP; NMS, SNMP; UNIX.

ABSTRACT
Snmpsrc is a network monitoring application that starts at a specified router in the network and traces the path of a given destination network from the starting router.

MECHANISM
Snmpsrc uses SNMP to gather its information. The starting router and destination network must be specified on the command line.

CAVEATS
An SNMP agent must be running on all of the routers in the path to the destination network in order for a complete path to be reported back to the user. The same SNMP community must also be configured in every SNMP agent in the path to the destination network.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
See CAVEATS.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY
Snmpsrc is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmpstat

KEYWORDS
manager, status; IP; NMS, SNMP; UNIX.

ABSTRACT
Snmpstat is a network monitoring application that gathers specific information from a network at regular intervals and stores this information into a commercial database. A report generation package is included in the NYSERNet SNMP Software Distribution to produce reports and graphs of raw data collected via SNMP.

MECHANISM
Snmpstat uses SNMP to gather its information. The agents which must be queried and the variables to query for are specified in a configuration file.

CAVEATS
An SNMP agent must be running in the network entity being monitored in order for snmpstat to be useful.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
SNMP polling is done synchronously. Currently, Ingres is the only commercial database supported. SQL is the query language being used.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY
Snmpstat is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmptrapd

KEYWORDS
alarm, manager; IP; NMS, SNMP; UNIX.

ABSTRACT
Snmptrapd is an SNMP trap agent that runs on UNIX derivatives. It receives and logs traps which are generated from snmp agents. A report generation package is included in the NYSERNet SNMP Software Distribution to produce reports and graphs of raw data collected via SNMP.

MECHANISM
Snmptrapd conforms to SNMP as specified in RFC 1098. Certain user configurable options are manipulated through a simple configuration file.

CAVEATS
None.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
Snmptrapd only logs traps into a UNIX flat file.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant.

AVAILABILITY
Snmptrapd is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmpwatch

KEYWORDS
manager, status; IP; NMS, SNMP; UNIX.

ABSTRACT
Snmpwatch is a network monitoring application that monitors variables in a single network entity and reports when they have changed value.

MECHANISM
Snmpwatch uses SNMP to gather its information. The entity to be monitored and the variables to be watched must be specified on the command line. Once a value changes, snmpwatch prints out the value and the variable to the standard output.

CAVEATS
An SNMP agent must be running on the network entity being monitored. Upon invocation, the initial value of each variable will printed out to the standard output.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
Only one network entity can be managed per invocation.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library.

AVAILABILITY
Snmpwatch is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmpxbar

KEYWORDS
manager, status; IP; NMS, SNMP, X; UNIX.

ABSTRACT
Snmpxbar is a network monitoring application based on X-Windows Version 11 Release 2 and the Simple Network Management Protocol. The application monitors a single numeric MIB object and displays its value in a bar chart. Snmpxbar supports color graphics.

MECHANISM
Snmpxbar uses SNMP to gather its information. The MIB object to be graphed must be specified on the command line. The polling interval can be changed dynamically from within snmpxbar.

CAVEATS
An SNMP agent must be running in the network entity being monitored in order for snmpxbar to be useful.

BUGS
Bugs are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
Can only graph one numeric MIB object per invocation.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library. X-Windows.

AVAILABILITY
Snmpxbar is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmpxconn

KEYWORDS
manager, map, status; IP; NMS, SNMP, X; UNIX.

ABSTRACT
Snmpxconn is a network monitoring application based on X-Windows Version 11 Release 2 and the Simple Network Management Protocol. The application monitors a number of (configurable) network entities and graphically depicts the TCP connections associated with the network entities via a TCP topology map.

MECHANISM
Snmpxconn uses SNMP to gather its information. A configuration file is used to determine the network entities to be monitored. There are certain command line arguments which manipulate the X environment and SNMP actions.

CAVEATS
An SNMP agent must be running in the network entity being monitored in order for snmpxconn to be useful.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
SNMP polling is done synchronously. The network entities must be configured by manually adding information to a configuration file.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library. X-Windows.

AVAILABILITY
Snmpxconn is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmpxmon

KEYWORDS

manager, map, status; IP; NMS, SNMP, X; UNIX.

ABSTRACT

Snmpxmon is a network monitoring application based on X-Windows Version 11 Release 2 and the Simple Network Management Protocol. This application will determine the status of sites and links it is configured to monitor (via its configuration file) by querying the designated sites and then displaying the result in a map form. Snmpxmon supports color graphics.

MECHANISM

Snmpxmon uses SNMP to gather its information. A configuration file is used to design the topology map. There are certain command line arguments which manipulate the X environment and SNMP actions.

CAVEATS

An SNMP agent must be running in the network entity being monitored in order for snmpxmon to be useful.

BUGS

None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS

SNMP polling is done synchronously. The topology map must be configured by hand.

HARDWARE REQUIRED

Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED

Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library. X-Windows.

AVAILABILITY

Snmpxmon is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

snmpxperf

KEYWORDS

manager, status; IP; NMS, SNMP, X; UNIX.

ABSTRACT

Snmpxperf is a network monitoring application based on X-Windows Version 11 Release 2 and the Simple Network Management Protocol. The application monitors a single numeric MIB object and displays its value in an EKG style histogram. Snmpxperf supports color graphics.

MECHANISM

Snmpxperf uses SNMP to gather its information. The MIB object to be graphed must be specified on the command line. The polling interval can be changed dynamically from within snmpxperf.

CAVEATS

An SNMP agent must be running in the network entity being monitored in order for snmpxperf to be useful.

BUGS

Auto-scaling sometimes doesn't downscale the EKG-graph enough on large spikes. This results in some of the graph running into the button boxes at the top of the window. Generally, Bugs are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS

Can only graph one numeric MIB object per invocation.

HARDWARE REQUIRED

Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED

Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library. X-Windows.

AVAILABILITY

Snmpxperf is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmpxperfmon

KEYWORDS
manager, status, traffic; IP; NMS, SNMP, X; UNIX.

ABSTRACT
Snmpxperfmon is a network monitoring application based on X-Windows Version 11 Release 2 and the Simple Network Management Protocol. The application monitors a single Network Entity and displays graphical information pertaining to the entities interface traffic statistics. Snmpxperfmon supports color graphics.

MECHANISM
Snmpxperfmon uses SNMP to gather its information. The MIB agent to be polled must be specified on the command line. The agent is then queried about all of its interfaces. Four EKG-style graphs are constructed for each interface (input pkts, output pkts, input Octets, output Octets).

CAVEATS
An SNMP agent must be running in the network entity being monitored in order for snmpxperfmon to be useful.

BUGS
Generally, bugs are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
Can only graph one network entity per invocation. Can only graph the amount of interfaces which will fit on a single bitmap display. Does not auto-scale or resize.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library. X-Windows.

AVAILABILITY
Snmpxperfmon is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME
snmpxrtmetric

KEYWORDS
manager, routing; IP; NMS, SNMP, X; UNIX.

ABSTRACT
Snmpxrtmetric is a network monitoring application based on X-Windows Version 11 Release 2 and the Simple Network Management Protocol. The application monitors the routing table of a specific agent and displays the RIP routing metric of certain destination networks in bar chart format.

MECHANISM
Snmpxrtmetric uses SNMP to gather its information. A configuration file is used to determine which destination networks will be graphed. The agent to be queried is specified on the command line.
Snmpxrtmetrtic supports color graphics.

CAVEATS
An SNMP agent must be running in the network entity being monitored in order for snmpxrtmetric to be useful.

BUGS
None outstanding. They are fixed as reports come in. Report bugs to: nysersnmp@nisc.nyser.net

LIMITATIONS
SNMP polling is done synchronously. The destination networks must be configured by manually adding information to a configuration file.

HARDWARE REQUIRED
Developed on Sun 3/60, Sun 3/260, tested on a SPARCstation I, DECstation, and a Solbourne 4/802.

SOFTWARE REQUIRED
Some UNIX variant or some other OS with a Berkeley Socket Compatibility Library. The X window system.

AVAILABILITY
Snmpxrtmetric is available in the NYSERNet SNMP Software Distribution, which is licensed, copyrighted software. To obtain information regarding the package send mail to: snmplisc@nisc.nyser.net or call +1 518-283-8860.

NAME

SpiderMonitor P220, K220 and
SpiderAnalyzer P320, K320

KEYWORDS

alarm, analyzer, generator, traffic; DECnet, ethernet, IP, OSI; eavesdrop; standalone; sourcelib.

ABSTRACT

The SpiderMonitor and SpiderAnalyzer are protocol analyzers for performing ethernet LAN diagnostics, monitoring, traffic generation, and troubleshooting. The SpiderMonitor has the capability of capturing every packet on a network and of decoding the first four layers of the OSI protocol model. The SpiderAnalyzer has additional software for decoding higher protocol layers. Protocol suites understood: TCP/IP (including SNMP and applications-layer protocols), OSI, XNS, DECnet and IPX. User-definable decodes can be written in 'C' with the Microsoft version 5.0 'C' compiler. A decode guide is provided.

The SpiderAnalyzer supports multiple simultaneous filters for capturing packets using predefined patterns and error states. Filter patterns can also trigger on NOT matching 1 or more filters, an alarm, or a specified time.

The SpiderAnalyzer can also employ TDR (Time Domain Reflectometry) to find media faults, open or short circuits, or transceiver faults. It can transmit OSI, XNS, and Xerox link-level echo packets to user-specified stations, performs loop round tests.

In traffic generation mode, the SpiderAnalyzer has the ability to generate packets at random intervals of random lengths or any combination of random or fixed interval or length, generation of packets with CRC errors, or packets that are too short, or packets that are too long.

Output from the SpiderMonitor/Analyzer can be imported to database or spreadsheet packages.

MECHANISM

The SpiderMonitor and Spider Analyzer are available as stand-alone, IBM PC compatible packages based upon a Compaq III portable system, or as a plug-in boards for any IBM XT/AT compatible machine. The model 220 (SpiderMonitor) systems provide a functional base suited for most network management needs. The model 320 (SpiderAnalyzer) systems provide extended functionality in the development mode and traffic generation mode as well more filtering capabilities than the 220 models.

CAVEATS

Traffic generation will congest an operational ethernet.

BUGS

None known.

LIMITATIONS

Monitoring of up to 1024 stations and buffering of up to 1500 packets. The model 220 provides for 3 filters with a filter depth of 46 bytes. The model 320 provides for 4 filters and a second level of filtering with a filter depth of 64 bytes.

HARDWARE REQUIRED

PX20s are self contained, the KX20s require an IBM PC/XT-AT compatible machine with 5 megabytes of hard disk storage and the spare slot into which the board kit is plugged.

SOFTWARE REQUIRED

None. The SpiderAnalyzer requires the Microsoft 'C' Compiler, Version 5.0 for writing user defined decodes.

AVAILABILITY

The SpiderMonitor/Analyzer is available commercially. For information on your local representative, call or write:

Spider Systems, Inc.
12 New England Executive Park
Burlington, MA 01803
Telephone: 617-270-3510
FAX: 617-270-9818

NAME

SPIMS — the Swedish Institute of Computer Science (SICS) Protocol Implementation Measurement System tool.

KEYWORDS

benchmark, debugger; IP, OSI; spoof; UNIX.

ABSTRACT

SPIMS is used to measure the performance of protocol and ‘‘protocol-like’’ services including response time (two-way delay), throughput and the time to open and close connections. It has been used to:

- benchmark alternative protocol implementations,
- observe how performance varies when parameters in specific implementations have been varied (i.e., to tune parameters).

SPIMS currently has interfaces to the DoD Internet Protocols: UDP, TCP, FTP, SunRPC, the OSI protocols from the ISODE 4.0 distribution package: FTAM, ROSE, ISO TP0 and to Sunlink 5.2 ISO TP4 as well as Stanford’s VMTP. Also available are a rudimentary set of benchmarks, stubs for new protocol interfaces and a user manual. For an example of the use of SPIMS to tune protocols, see:

> Nordmark & Cheriton, ‘‘Experiences from VMTP: How to achieve low response time,’’ *IFIP WG6.1/6.4: Protocols for High-Speed Networks,* May 1989, Zurich. To be published.

MECHANISM

SPIMS runs as user processes and uses a TCP connection for measurement set-up. Measurements take place between processes over the measured protocol. SPIMS generates messages and transfers them via the measured protocol service according to a user-supplied specification. SPIMS has a unique measurement specification language that is used to specify a measurement session. In the language there are constructs for different application types (e.g., bulk data transfer), for specifying frequency and sequence of messages, for distribution over message sizes and for combining basic specifications. These specifications are independent of both protocols and protocol implementations and can be used for benchmarking. For more details on the internals of SPIMS, see:

> Nordmark & Gunningberg, ‘‘SPIMS: A Tool for Protocol Implementation Performance Measurements’’ *Proc. of 13:th Conf. on Local Computer Networks,* Minneapolis 1989, pp 222-229.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

SPIMS is implemented on UNIX, including SunOS 4., 4.3BSD UNIX, DN (UNIX System V, with extensions) and Ultrix 2.0/3.0. It requires a TCP connection for measurement set-up. No kernel modifications or any modifications to measured protocols are required.

AVAILABILITY

SPIMS is not in the public domain; the software is covered by licenses. The Swedish Institute of Computer Science has released the research prototype of SPIMS for research and non-commercial use. Commercial organizations may obtain the research prototype, but it is for internal research only and for no commercial use whatsoever. A commercial, supported version of SPIMS is distributed by TeleLOGIC Uppsala AB, Sweden.

For universities and non-profit organizations, SPIMS source code is distributed free of charge. There are two ways to get the software:

1. FTP. If you have an Internet FTP connection, you can use anonymous FTP to sics.se [192.16.123.90], and retrieve the file in pub/spims-dist/dist890915.tar.Z (this is a .6MB tar image) in BINARY mode. Log in as user anonymous and at the password prompt, use your complete electronic mail address.
2. On a Sun 1/4-inch cartridge tape. For mailing, a handling fee of US$150.00 will be charged. Submit a bank check with the request. Do not send tapes or envelopes.

For other organizations, the SPIMS source code for the research prototype is distributed for a one-time fee of US$500.00. Organizations interested in the research prototype need to contact SICS via email and briefly motivate why they qualify (non-commercial use) for the research prototype. They will thereafter get a permission to obtain a copy from the same distribution source as for universities.

For more information about the research prototype distribution, contact:

Swedish Institute of Computer Science
Att: Birgitta Klingenberg
P.O. Box 1263
S-164 28 Kista
SWEDEN

e-address: spims@sics.se
Phone: +46-8-7521500, Fax: +46-8-7517230

TeleLOGIC Uppsala AB, a subsidiary of Swedish Telecom, distributes and supports a version of SPIMS for commercial use. It consists of object code for SunOS 4., 4.3BSD UNIX, DNIX, and Ultrix 2.0/3.0. Support for other UNIX-like implementations will be considered according to demand. The same interfaces to the DoD Internet and OSI protocols from the ISODE 4.0 are included as well as a user manual.

For further information about SPIMS for the commercial user please contact:

Claes Hojenberg
TeleLOGIC Uppsala AB
P.O. Box 1218
S-751 42 UPPSALA
Sweden

e-address: claes@uplog.se
Phone: +46-18-189400, Fax: +46-18-132039

NAME

spray

KEYWORDS

benchmark, generator; IP; ping; UNIX.

ABSTRACT

Spray is a traffic generation tool that generates RPC or UDP packets, or ICMP Echo Requests. The packets are sent to a remote procedure call application at the destination host. The count of received packets is retrieved from the remote application after a certain number of packets have been transmitted. The difference in packets received versus packets sent represents (on a LAN) the packets that the destination host had to drop due to increasing queue length. A measure of throughput relative to system speed and network load can thus be obtained.

MECHANISM

See above.

CAVEATS

Spray can congest a network.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

SunOS

AVAILABILITY

Supplied with SunOS.

NAME
: tcpdump

KEYWORDS
: traffic; ethernet, IP, NFS; UNIX, VMS; free.

ABSTRACT
: Tcpdump can interpret and print headers for the following protocols: ethernet, IP, ICMP, TCP, UDP, NFS, ND, ARP/RARP, AppleTalk. Tcpdump has proven useful for examining and evaluating the retransmission and window management operations of TCP implementations.

MECHANISM
: Much like etherfind, tcpdump writes a log file of the frames traversing an ethernet interface. Each output line includes the time a packet is received, the type of packet, and various values from its header.

CAVEATS
: None.

BUGS
: None known.

LIMITATIONS
: Public domain version requires a kernel patch for SunOS.

HARDWARE REQUIRED
: Ethernet.

SOFTWARE REQUIRED
: BSD UNIX or related OS, or VMS.

AVAILABILITY
: Available, though subject to copyright restrictions, via anonymous FTP from ftp.ee.lbl.gov. The source and documentation for the tool is in compressed tar format, in file tcpdump.tar.Z. Also available from spam.itstd.sri.com, in directory pub. For VMS hosts with DEC ethernet controllers, available as part of TGV MultiNet IP software package.

NAME
: tcplogger

KEYWORDS
: traffic; IP; eavesdrop; UNIX; free.

ABSTRACT
: Tcplogger consists of modifications to the 4.3BSD UNIX source code, and a large library of post-processing software. Tcplogger records timestamped information from TCP and IP packets that are sent and received on a specified connection. For each TCP packet, information such as sequence number, acknowledgement sequence number, packet size, and header flags is recorded. For an IP packet, header length, packet length and TTL values are recorded. Customized use of the TCP option field allows the detection of lost or duplicate packets.

MECHANISM
: Routines of 4.3BSD UNIX in the netinet directory have been modified to append information to a log in memory. The log is read continuously by a user process and written to a file. A TCP option has been added to start the logging of a connection. Lots of post-processing software has been written to analyze the data.

CAVEATS
: None.

BUGS
: None known.

LIMITATIONS
: To get a log at both ends of the connection, the modified kernel should be run at both the hosts.

  All connections are logged in a single file, but software is provided to filter out the record of a single connection.

HARDWARE REQUIRED
: No restrictions.

SOFTWARE REQUIRED
: 4.3BSD UNIX (as modified for this tool).

AVAILABILITY
: Free, although a 4.3BSD license is required. Contact Olafur Gudmundsson (ogud@cs.umd.edu).

NAME
TokenVIEW

KEYWORDS
control, manager, status; ring; NMS, proprietary; DOS.

ABSTRACT
Network Management tool for 4/16 Mbit IEEE 802.5 Token Ring Networks. Monitors active nodes and ring errors. Maintains database of nodes, wire centers and their connections. Separate network management ring allows remote configuration of wire centers.

MECHANISM
A separate network management ring used with Proteon Intelligent Wire Centers allows wire center configuration information to be read and modified from a single remote workstation. A log of network events used with a database contain nodes, wire centers and their connections, facilitates tracking and correction of network errors. Requires an ‘‘E’’ series PROM, sold with package.

CAVEATS
Currently, only ISA bus cards support the required E series PROM.

BUGS
None known.

LIMITATIONS
256 nodes, 1 net.

HARDWARE REQUIRED
512K RAM, CGA or better, hard disk, mouse supported.

SOFTWARE REQUIRED
MS-DOS, optional mouse driver

AVAILABILITY
Fully supported product of Proteon, Inc. Previously sold as Advanced Network Manager (ANM). For more information, contact:

Proteon, Inc.	Phone: (508) 898-2800
2 Technology Drive	Fax: (508) 366-8901
Westborough, MA 01581	Telex: 928124

NAME

traceroute

KEYWORDS

routing; IP; ping; UNIX, VMS; free.

ABSTRACT

Traceroute is a tool that allows the route taken by packets from source to destination to be discovered. It can be used for situations where the IP record route option would fail, such as intermediate gateways discarding packets, routes that exceed the capacity of an datagram, or intermediate IP implementations that don't support record route. Round trip delays between the source and intermediate gateways are also reported allowing the determination of individual gateways contribution to end-to-end delay.

Enhanced versions of traceroute have been developed that allow specification of loose source routes for datagrams. This allows one to investigate the return path from remote machines back to the local host.

MECHANISM

Traceroute relies on the ICMP TIME_EXCEEDED error reporting mechanism. When an IP packet is received by an gateway with a time-to-live value of 0, an ICMP packet is sent to the host which generated the packet. By sending packets to a destination with a TTL of 0, the next hop can be identified as the source of the ICMP TIME EXCEEDED message. By incrementing the TTL field the subsequent hops can be identified. Each packet sent out is also time stamped. The time stamp is returned as part of the ICMP packet so a round trip delay can be calculated.

CAVEATS

Some IP implementations forward packets with a TTL of 0, thus escaping identification. Others use the TTL field in the arriving packet as the TTL for the ICMP error reply, which delays identification.

Sending datagrams with the source route option will cause some gateways to crash. It is considered poor form to repeat this behavior.

BUGS

None known.

LIMITATIONS

Most versions of UNIX have errors in the raw IP code that require kernel mods for the standard version of traceroute to work. A version of traceroute exists that runs without kernel mods under SunOS 3.5 (see below), but it only operates over an ethernet interface.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

BSD UNIX or related OS, or VMS.

AVAILABILITY

Available by anonymous FTP from ftp.ee.lbl.gov, in file traceroute.tar.Z. It is also available from uc.msc.umn.edu.

A version of traceroute that supports Loose Source Record Route, along with the source code of the required kernel modifications and a Makefile for installing them, is available via anonymous FTP from zerkalo.harvard.edu, in directory pub, file traceroute_pkg.tar.Z.

A version of traceroute that runs under SunOS 3.5 and does NOT require kernel mods is available via anonymous FTP from dopey.cs.unc.edu, in file ~ftp/pub/traceroute.tar.Z.

For VMS, traceroute is available as part of TGV MultiNet IP software package.

NAME

TRPT — transliterate protocol trace

KEYWORDS

traffic; IP; eavesdrop; UNIX; free.

ABSTRACT

TRPT displays a trace of a TCP socket events. When no options are supplied, TRPT prints all the trace records found in a system, grouped according to TCP connection protocol control block (PCB).

An example of TRPT output is:

```
38241 ESTABLISHED:input [e0531003..e0531203)@6cc5b402(win=4000)<ACK> -> ESTABLISHED
38241 ESTABLISHED:user RCVD -> ESTABLISHED
38266 ESTABLISHED:output 6cc5b402@e0531203(win=4000)<ACK> -> ESTABLISHED
38331 ESTABLISHED:input [e0531203..e0531403)@6cc5b402(win=4000)<ACK,FIN,PUSH> ->
CLOSE_WAIT
38331 CLOSE_WAIT:output 6cc5b402@e0531404(win=3dff)<ACK> -> CLOSE_WAIT
38331 CLOSE_WAIT:user RCVD -> CLOSE_WAIT
38343 LAST_ACK:output 6cc5b402@e0531404(win=4000)<ACK,FIN> -> LAST_ACK
38343 CLOSE_WAIT:user DISCONNECT -> LAST_ACK
38343 LAST_ACK:user DETACH -> LAST_ACK
```

MECHANISM

TRPT interrogates the buffer of TCP trace records that is created when a TCP socket is marked for debugging.

CAVEATS

Prior to using TRPT, an analyst should take steps to isolate the problem connection and find the address of its protocol control blocks.

BUGS

None reported.

LIMITATIONS

A socket must have the debugging option set for TRPT to operate. Another problem is that the output format of TRPT is difficult.

HARDWARE REQUIRED

No restrictions.

SOFTWARE REQUIRED

BSD UNIX or related OS.

AVAILABILITY

Included with BSD and SunOS distributions. Available via anonymous FTP from uunet.uu.net, in file bsd-sources/src/etc/trpt.tar.Z.

NAME
TTCP

KEYWORDS
benchmark, generator; IP; ping; UNIX, VMS; free.

ABSTRACT
TTCP is a traffic generator that can be used for testing end-to-end throughput. It is good for evaluating TCP/IP implementations.

MECHANISM
Cooperating processes are started on two hosts. The open a TCP connection and transfer a high volume of data. Delay and throughput are calculated.

CAVEATS
Will greatly increase system load.

BUGS
None known.

LIMITATIONS
None reported.

HARDWARE REQUIRED
No restrictions.

SOFTWARE REQUIRED
BSD UNIX or related OS, or VMS.

AVAILABILITY
Source for BSD UNIX is available via anonymous FTP from vgr.brl.mil, in file ftp/pub/ttcp.c, and from sgi.com, in file sgi/src/ttcp.c. A version of TTCP has also been submitted to the USENET news group comp.sources.unix. For VMS, ttcp.c is included in the MultiNet Programmer's Kit, a standard feature of TGV MultiNet IP software package.

NAME

Unisys Network Control Center (NCC)

KEYWORDS

alarm, benchmark, control, generator, manager, map, reference, status, traffic; ethernet, FDDI, IP; NMS, ping, SNMP; UNIX.

ABSTRACT

The Unisys Defense Systems Network Control Center (NCC) provides high-performance software to support the management and control of TCP/IP-based networks. The network management system uses the Simple Network Management Protocol (SNMP) to exchange management information between the NCC and network devices. The NCC supports the Management Information Base (MIB) [RFC-1066] and the Structure and Identification of Management Information for TCP/IP-based Internets [RFC-1065]. In addition, Unisys has extended the MIB definitions to support the features of Unisys FDDI LAN devices, such as the FDDI Smart Concentrators, the FDDI Host Network Front Ends, and the Remote FDDI, FDDI-to-LAN, and FDDI-to-DDN gateways.

The NCC supports seven applications. The network topology map displays the physical and logical maps of the network. The configuration management tool supports the modification and validation of network device configuration data as well as the modification of MIB configuration data. The performance monitoring tool supports the collection and analysis of statistical parameters from network devices. The status monitoring tool reports on the up/down status and responsiveness of network devices using ICMP. The accounting tool is used to collect, store, and display user job activity at the subscriber hosts. The NCC database entry supports RFC 1066 object definitions and Unisys-specific object definitions to support the Unisys FDDI devices. And finally, the trap reporting tool reports the arrival of error and event notifications using UDP datagrams. The NCC supports all the trap messages defined in RFC 1098.

MECHANISM

The NCC is based on the Simple Network Management Protocol (SNMP).

CAVEATS

None.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

A minimal platform consists of a Sun 3/60FC-8, with at least 200 MB disk and cartridge tape (1/4"). A full-sized color monitor, more disk, and a workstation based on a higher performance processor is beneficial to NCC activities.

SOFTWARE REQUIRED

SunOS Version 4.0 running the SunView windowing environment and the SYBASE Relational Data Base Management System.

AVAILABILITY

Commercially available as a turn-key package or as a software product from:

Unisys Defense Systems
5151 Camino Ruiz
Camarillo, California 93010
(805) 987-6811
(Dale Russell <dsr@cam.unisys.com>)

NAME

WIN/MGT Station — Network Management Station for SunOS.

KEYWORDS

alarm, control, manager, routing, status, traffic; ethernet, IP; NMS, SNMP, X; UNIX; library.

ABSTRACT

WIN/MGT Station for SunOS is a network management software product based on the SNMP. It provides the capability to manage standards-based networking products from The Wollongong Group as well as other vendors. Fully compliant with RFCs 1065, 1066 and 1098, WIN/MGT Station uses a menu-driven graphical user interface.

WIN/MGT capabilities include configuration, performance and fault management for SNMP-based agents. The WIN/MGT station can perform polling to monitor the status of all MIB variables defined in RFC 1066, ‘‘Management Information Base for network management of TCP/IP-based internets.’’ In addition, the WIN/MGT Station can process ‘‘trap’’ messages from SNMP agents. Furthermore, the WIN/MGT Station can support any private extension to the Management Information Base with minimal user configuration.

An icon-driven network interface map allows the user to monitor their network topology and status. Changes in the operational status of any manageable network element is displayed visually and audibly.

The WIN/MGT package includes an Applications Programming Interface (API) for the ‘‘C’’ language. The API is a set of libraries that enable an applications program to perform SNMP ‘‘set’’ and ‘‘get’’ operations. This allows users to integrate site-specific applications with WIN/MGT.

SNMP agent software for the Sun 3 host is also provided so that the Network Management Station itself can also be monitored and managed.

MECHANISM

The WIN/MGT Station uses SNMP to monitor and control SNMP agents.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

A theoretical limitation of approximately 18,000 network elements can be managed.

HARDWARE REQUIRED

Any model of Sun 3 system. Recommended minimums include 8 MB RAM, 100 MB disk space (30 MB to start), and color monitor. Also tested on DECstation 3100, PS/2 (with SCO UNIX) and Macintosh IIcx computer using A/UX.

SOFTWARE REQUIRED

SunOS 4.x. MIT X Window System, Release 11, version 3, or OpenWindows (X.11/NeWS) from Sun Microsystems, Inc. WIN/MGT Station for SunOS is provided on 1/4" tape in cpio format.

AVAILABILITY

A commercial product of:

The Wollongong Group, Inc.
1129 San Antonio Rd.
Palo Alto, CA 94303
(415) 962-7200 br fax (415) 968-3619
internet oldera@twg.com

NAME

xnetmon, xpmon

KEYWORDS

alarm, manager, map, status; IP; NMS, SNMP; UNIX.

ABSTRACT

Xnetmon and xpmon provide graphical representation of performance and status of SNMP-capable network elements. Xnetmon presents a schematic network map representing the up/down status of network elements; xpmon draws a pen plot style graph of the change over time of any arbitrary MIB object (RFC1066). Both xnetmon and xpmon use the SNMP (RFC1098) for retrieving status and performance data.

MECHANISM

Xnetmon polls network elements for the status of their interfaces on a controllable polling interval. Pop-up windows displaying the values of any MIB variable are supported by separate polls. When SNMP traps are received from a network element, that element and all adjacent elements are immediately re-polled to update their status. The layout of the network map is statically configured. Xpmon repeatedly polls (using SNMP) the designated network element for the value of the designated MIB variable on the user-specified interval. The change in the variable is then plotted on the strip chart. The strip chart regularly adjusts its scale to the current maximum value on the graph.

CAVEATS

Polling intervals should be chosen with care so as not to affect system performance adversely.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

Distributed and supported for Sun-3 systems.

SOFTWARE REQUIRED

SunOS 3.5 or 4.x; X11, release 2 or 3.

AVAILABILITY

Commercial product of:

Wellfleet Communications, Inc.
12 DeAngelo Drive
Bedford, MA 01730-2204
(617) 275-2400

NAME

XNETMON — an X windows based SNMP network management station from SNMP Research.

KEYWORDS

alarm, control, manager, map, routing, security, status; DECnet, ethernet, IP, OSI, ring, star; NMS, SNMP, X; DOS, UNIX, VMS; sourcelib.

ABSTRACT

The XNETMON application implements a powerful network management station based on the X window system. It provides network managers tools for fault management, configuration management, performance management, and security management. It can be successfully used with many types of networks including those based on various LAN media, and wide area networks. XNETMON has been used with multiprotocol devices including those which support TCP/IP, DECnet, and OSI protocols. The fault management tool displays the map of the network configuration with node and link state indicated in one of several colors to indicate current status. Alarms may be enabled to alert the operator of events occurring in the network. Events are logged to disk. The configuration management tool may be used to edit the network management information base stored in the network management station to reflect changes occurring in the network. Other features include graphs and tabular tools for use in fault and performance management and mechanisms by which additional variables, such as vendor-specific variables, may be added. The XNETMON application comes complete with source code including a powerful set of portable libraries for generating and parsing SNMP messages. Output data from XNETMON may be transferred via flat files for additional report generation by a variety of statistical packages.

MECHANISM

The XNETMON application is based on the Simple Network Management Protocol (SNMP). Polling is performed via the powerful SNMP get-next operator and the SNMP get operator. Trap directed polling is used to regulate the focus and intensity of the polling.

CAVEATS

None.

BUGS

None known.

LIMITATIONS

The monitored and managed nodes must implement the SNMP over UDP per RFC 1098 or must be reachable via a proxy agent.

HARDWARE REQUIRED

X windows workstation with UDP socket library. Monochrome is acceptable but color is far superior.

SOFTWARE REQUIRED

X windows version 11 release 3 or later.

AVAILABILITY

This is a commercial product available under license from:

SNMP Research
P.O. Box 8593
Knoxville, TN 37996-4800
(615) 573-1434 (Voice)
(615) 573-9197 (FAX)
Attn: Dr. Jeff Case

NAME

xnetperfmon — a graphical network performance and fault management tool from SNMP Research.

KEYWORDS

manager, status; DECnet, ethernet, IP, OSI, ring, star; NMS, SNMP, X; DOS, UNIX, VMS; sourcelib.

ABSTRACT

Xnetperfmon may be used to plot SNMP variables as a graphical display. These graphs are often useful for fault and performance management. Variables may be plotted as gauges versus time. Alternatively, counters may be plotted as delta count/delta time (rates). The user may easily customize the variables to be plotted, labels, step size, update interval, and the like. The scales automatically adjust whenever a point to be plotted would go off scale.

MECHANISM

The xnetperfmon application communicates with remote agents or proxy agents via the Simple Network Management Protocol (SNMP).

CAVEATS

All plots for a single invocation of xnetperfmon must be for variables provided by a single network management agent. However, multiple invocations of xnetperfmon may be active on a single display simultaneously or proxy agents may be used to summarize information at a common point.

BUGS

None known.

LIMITATIONS

None reported.

HARDWARE REQUIRED

Systems supporting X windows.

SOFTWARE REQUIRED

X Version 11 release 2 or later.

AVAILABILITY

This is a commercial product available under license from:

SNMP Research
P.O. Box 8593
Knoxville, TN 37996-4800
(615) 573-1434 (Voice)
(615) 573-9197 (FAX)
Attn: Dr. Jeff Case

NAME
xup

KEYWORDS
status; ping, X; HP.

ABSTRACT
Xup uses the X-Windows to display the status of an ‘‘interesting’’ set of hosts.

MECHANISM
Xup uses ping to determine host status.

CAVEATS
Polling for status increases network load.

BUGS
None known.

LIMITATIONS
None reported.

HARDWARE REQUIRED
Runs only on HP series 300 and 800 workstations.

SOFTWARE REQUIRED
Version 10 of X-Windows.

AVAILABILITY
A standard command for the HP 300 & 800 Workstations.

Network Management Tutorial

This tutorial is an overview of the practice of network management. Reading this section is no substitute for knowing your system, and knowing how it is used. Do not wait until things break to learn what they ought to do or how they usually work: a crisis is not the time for determining how ''normal'' packet traces should look. Furthermore, it takes little imagination to realize that you do not want to be digging through manuals while your boss is screaming for network service to be restored.

We assume an acquaintance with the TCP/IP protocol suite and the Internet architecture. There are many available references on these topics, several of which are listed below in Section 7.

Since many of the details of network management are system-specific, this tutorial is a bit superficial. There is, however, a more fundamental problem in prescribing network management practices: network management is not a well-understood endeavor. At present, the cutting edge of network management is the use of distributed systems to collect and exchange status information, and then to display the data as histograms or trend lines. It is not clear that we know what data should be collected, how to analyze it when we get it, or how to structure our collection systems. For now, automated, real-time control of internets is an aspiration, rather than a reality. The communications systems that we field are apparently more complex than we can comprehend, which no doubt accounts in part for their frequently surprising behavior.

The first section of this tutorial lists the overall goals and functions of network management. It presents several aspects of network management, including system monitoring, fault detection and isolation, performance testing, configuration management, and security. These discussions are followed by a bibliographic section. The tutorial closes with some final advice for network managers.

1. Network Management Goals and Functions

An organization's view of network management goals is shaped by two factors:

1. people in the organization depend on the system working,
2. LANs, routers, lines, and other communications resources have costs.

From the organizational vantage point, the ultimate goal of network management is to provide a consistent, predictable, acceptable level of service from the available data communications resources. To achieve this, a network manager must first be able to perform fault detection, isolation, and correction. He must also be able to effect configuration changes with a minimum of disruption, and measure the utilization of system components.

People actually managing networks have a different focus. Network managers are usually evaluated by the availability and performance of their communications systems, even though many factors of net performance are beyond their control. To them, the most important requirement of a network management tool is that it allows the detection and diagnosis of faults before users can call to complain: users (and bosses) can often be placated just by knowing that a network problem has been diagnosed. Another vital network management function is the ability to collect data that justify current or future expenditures for the data communications plant and staff.

Following a section on system monitoring, this tutorial addresses fault, performance, configuration, and security management. By fault management, we mean the detection, diagnosis, and correction of network malfunctions. Under the subject of performance management, we include support for predictable, efficient service, as well as capacity planning and capacity testing. Configuration management includes support for orderly configuration changes (usually, system growth), and local administration of component names and addresses. Security management includes both protecting system components from damage and protecting sensitive information from unintentional or malicious disclosure or corruption.

Readers familiar with the ISO management standards and drafts will note both that we have borrowed heavily from the ''OSI Management Framework,'' except that we have omitted the ''account management'' function. Account management seems a bit out of place with the other network management functions. The logging required by account management is likely to be done by specialized, dedicated subsystems that are distinct from other network management components. Hence, this tutorial does not cover account management. Rest assured, however, that account management, if required, will be adequately supported and staffed.

For those with a DoD background, security may also seem out of place as a subtopic of network management. Without doubt, communications security is an important issue that should be considered in its own right. Because of the requirements of trust for security mechanisms, security components will probably not be integrated subcomponents of a larger network management system. Nevertheless, because a network manager has a responsibility to protect his system from undue security risks, this tutorial includes a discussion on internet security.

2. System Monitoring

System monitoring is a fundamental aspect of network management. One can divide system monitoring into two rough categories: error detection and baseline monitoring.

System errors, such as misformatted frames or dropped packets, are not in themselves cause for concern. Spikes in error rates, however, should be investigated. It is sound practice to log error rates over time, so that increases can be recognized. Furthermore, logging error rates as a function of traffic rates can be used to detect congestion. Investigate unusual error rates and other anomalies as they are detected, and keep a notebook to record your discoveries.

Day-to-day traffic should be monitored, so that the operational baselines of a system and its components can be determined. As well as being essential for performance management, baseline determination and traffic monitoring are the keys to early fault detection.

A preliminary step to developing baseline measurements is construction of a system map: a graphical representation of the system components and their interfaces. Then, measurements of utilization (i.e., use divided by capacity) are needed. Problems are most likely to arise, and system tuning efforts are most likely to be beneficial, at highly utilized components.

It is worthwhile to develop a source/destination traffic matrix, including a breakdown of traffic between the local system and other internet sites. Both volume and type of traffic should be logged, along with its evolution over time. Of particular interest for systems with diskless workstations is memory swapping and other disk server access. For all systems, broadcast traffic and routing traffic should be monitored. Sudden increases in the variance of delay or the volume of routing traffic may indicate thrashing or other soft failures.

In monitoring a system, long-term averages are of little use. Hourly averages are a better indicator of system use. Variance in utilization and delay should also be tracked. Sudden spikes in variance are tell-tale signs that a problem is looming or exists. So, too, are trends of increased packet or line errors, broadcasts, routing traffic, or delay.

3. Fault Detection and Isolation

When a system fails, caution is in order. A net manager should make an attempt to diagnose the cause of a system crash before rebooting. In many cases, however, a quick diagnosis will not be possible. For some high priority applications, restoring at least some level of service will have priority over fault repair or even complete fault diagnosis. This necessitates prior planning. A net manager must know the vital applications at his site. If applications require it, he must also have a fall-back plan for bringing them online. Meanwhile, repeated crashes or hardware failures are unambiguous signs of a problem that must be corrected.

A network manager should prepare for fault diagnosis by becoming familiar with how diagnostic tools respond to network failure. In times of relative peace, a net manager should occasionally unplug the network connection from an unused workstation and then ‘‘debug’’ the problem.

When diagnosing a fault or anomaly, it is vital to proceed in an orderly manner, especially since network faults will usually generate spurious as well as accurate error messages. Remember to keep in mind that the network itself is failing. Do not place too much trust in anything obtained remotely. Furthermore, it is unlikely to be significant that remote information such as DNS names or NFS files cannot be obtained.

Even spurious messages can be revealing, because they provide clues to the problem. From the data at hand, develop working hypotheses about probable causes of the problems you detect. Direct your further data gathering efforts so that the information you get will either refute or support your hypotheses.

An orderly approach to debugging is facilitated if it is guided by a model of network behavior. The following portions of this section present such a model, along with a procedure for checking network connectivity. The section concludes with some hints for diagnosing a particularly tricky class of connectivity problem.

3.1 A Network Model as a Diagnostic Framework

The point of having a model of how things work is to have a basis for developing educated guesses about how things go wrong. The problem of cascading faults — faults generating other faults — makes use of a conceptual model a virtual necessity.

In general, only problems in a component's hardware or operating system will generate simultaneous faults in multiple protocol layers. Otherwise, faults will propagate vertically (up the protocol stack) or horizontally (between peer-level communications components). Applying a conceptual model that includes the architectural relations of network components can help to order an otherwise senseless barrage of error messages and symptoms.

The model does not have to be formal or complex to bring structure to debugging efforts. A useful start is something as simple as the following:

1. Applications programs use transport services: TCP/UDP. Before using service, applications that accept host names as parameters must translate the names into IP addresses. Translation may be based on a static table lookup (/etc/hosts file in UNIX hosts), the DNS, or yellow pages. Nslookup and DiG are tools for monitoring the activities of the DNS.
2. Transport protocol implementations use IP services. The local IP module makes the initial decision on forwarding. An IP datagram is forwarded directly to the destination host if the destination is on the same network as the source. Otherwise, the datagram is forwarded to a gateway attached to the network. On BSD hosts, the contents of a host's routing table are visible by use of the ‘‘netstat’’ command.*
3. IP implementations translate the IP address of a datagram's next hop (either the destination host or a gateway) to a local network address. For ethernets, the Address Resolution Protocol (ARP) is commonly used for this translation. On BSD systems, an interface's IP address and other configuration options can be viewed by use of the ‘‘ifconfig’’ command, while the contents of a host's ARP cache may be viewed by use of ‘‘arp’’ command.
4. IP implementations in hosts and gateways route datagrams based on subnet and net identifiers. Subnetting is a means of allocating and preserving IP address space, and of insulating users from the topological details of a multi-network campus. Sites that use subnetting reserve portions of the IP address's host identifier to indicate particular networks at their campus. Subnetting is highly system-dependent. The details are a critical, though local, issue. As for routing between separate networks, a variety of gateway-to-gateway protocols are used. Traceroute is a useful tool for investigating routing problems. The tool, ‘‘query,’’ can be used to examine RIP routing tables.

---

* Initial forwarding may actually be complex and vulnerable to multiple points of failure. For example, when sending an IP datagram, 4.3BSD hosts first look for a route to the particular host. If none has been specified for the destination, then a search is made for a route to the network of the destination. If this search also fails, then as a last resort, a search is made for a route to a ‘‘default’’ gateway. Routes to hosts, networks, and the ‘‘default’’ gateway may be static, loaded at boot time and perhaps updated by operator commands. Alternatively, they may be dynamic, loaded from redirects and routing protocol updates.

A neophyte network manager should expand the above description so that it accurately describes his particular system, and learn the tools and techniques for monitoring the operations at each of the above stages.

3.2 A Simple Procedure for Connectivity Check

In this section, we describe a procedure for isolating a TCP/IP connectivity problem.** In this procedure, a series of tests methodically examine connectivity from a host, starting with nearby resources and working outward. The steps in our connectivity-testing procedure are:

1. As an initial sanity check, ping your own IP address and the loopback address.
2. Next, try to ping other IP hosts on the local subnet. Use numeric addresses when starting off, since this eliminates the name resolvers and host tables as potential sources of problems. The lack of an answer may indicate either that the destination host did not respond to ARP (if it is used on your LAN), or that a datagram was forwarded (and hence, the destination IP address was resolved to a local media address) but that no ICMP Echo Reply was received. This could indicate a length-related problem, or misconfigured IP Security.
3. If an IP router (gateway) is in the system, ping both its near and far-side addresses.
4. Make sure that your local host recognizes the gateway as a relay. (For BSD hosts, use netstat.)
5. Still using numeric IP addresses, try to ping hosts beyond the gateway. If you get no response, run hopcheck or traceroute, if available. Note whether your packets even go to the gateway on their way to the destination. If not, examine the methods used to instruct your host to use this gateway to reach the specified destination net (e.g., is the default route in place? Alternatively, are you successfully wiretapping the IGP messages broadcast on the net you are attached to?)

   If traceroute is not available, ping, netstat, arp, and a knowledge of the IP addresses of all the gateway's interfaces can be used to isolate the cause of the problem. Use netstat to determine your next hop to the destination. Ping that IP address to ensure the router is up. Next, ping the router interface on the far subnet. If the router returns ‘‘network unreachable’’ or other errors, investigate the router's routing tables and interface status. If the pings succeed, ping the close interface of the succeeding next hop gateway, and so on. Remember the routing along the outbound and return paths may be different.
6. Once ping is working with numeric addresses, use ping to try to reach a few remote hosts by name. If ping fails when host names are used, check the operation of the local name-mapping system (i.e., with nslookup or DiG). If you want to use ‘‘shorthand’’ forms (‘‘myhost’’ instead of ‘‘myhost.mydomain.com’’), be sure that the alias tables are correctly configured.
7. Once basic reachability has been established with ping, try some TCP-based applications: FTP and TELNET are supported on almost all IP hosts, but FINGER is a simpler protocol. The Berkeley-specific protocols (RSH, RCP, REXEC and LPR) require extra configuration on the server host before they can work, and so are poor choices for connectivity testing.

If problems arise in steps 2-7 above, rerunning the tests while executing a line monitor (e.g., etherfind, netwatch, or tcpdump) can help to pinpoint the problem.

The above procedure is sound and useful, especially if little is known about the cause of the connectivity problem. It is not, however, guaranteed to be the shortest path to diagnosis. In some cases, a binary search on the problem might be more effective (i.e., try a test ‘‘in the middle,’’ in a spot where the failure modes are well defined). In other cases, available information might so strongly suggest a particular failure that immediately testing for it is in order. This last ‘‘approach,’’ which might be called ‘‘hunting and pecking,’’ should be used with caution: chasing one will o' the wisp after another can waste much time and effort.

** Thanks to James VanBokkelen, president of FTP Software, for sharing with us a portion of a PC/TCP support document, the basis for the above connectivity procedure.

Note that line problems are still among the most common causes of connectivity loss. Problems in transmission across local media are outside the scope of this tutorial. But, if a host or workstation loses or cannot establish connectivity, check its physical connection.

3.3 Limited Connectivity

An interesting class of problems can result in a particularly mysterious failure: TELNET or other low-volume TCP connections work, but large file transfers fail. FTP transfers may start, but then hang. There are several possible culprits in this problem. The most likely suspects are IP implementations that cannot fragment or reassemble datagrams, and TCP implementations that do not perform dynamic window sizing (a.k.a. Van Jacobson's ''Slow Start'' algorithm). Another possibility is mixing incompatible frame formats on an ethernet.

Even today, some IP implementations in the Internet cannot correctly handle fragmentation or reassembly. They will work fine for small packets, but drop all large packets.

The problem can also be caused by buffer exhaustion at gateways that connect interfaces of widely differing bandwidth. Datagrams from a TCP connection that traverses a bottleneck will experience queue delays, and will be dropped if buffer resources are depleted. The congestion can be made worse if the TCP implementation at the traffic source does not use the recommended algorithms for computing retransmission times, since spuriously retransmitted datagrams will only add to the congestion.* Fragmentation, even if correctly implemented, will compound this problem, since processing delays and congestion will be increased at the bottleneck.

Serial Line Internet Protocol (SLIP) links are especially vulnerable to this and other congestion problems. SLIP lines are typically an order of magnitude slower than other gateway interfaces. Also, the SLIP lines are at times configured with MTUs (Maximum Transfer Unit, the maximum length of an IP datagram for a particular subnet) as small as 256 bytes, which virtually guarantees fragmentation.

To alleviate this problem, TCP implementations behind slow lines should advertise small windows. Also, if possible, SLIP lines should be configured with an MTU no less than 576 bytes. The tradeoff to weigh is whether interactive traffic will be penalized too severly by transmission delays of lengthy datagrams from concurrent file transfers.

Misuse of ethernet trailers can also cause the problem of hanging file transfers. ''Trailers'' refers to an ethernet frame format optionally employed by BSD systems to minimize buffer copying by system software. BSD systems with ethernet interfaces can be configured to send large frames so that their address and control data are at the end of a frame (hence, a ''trailer'' instead of a ''header''). After a memory page is allocated and loaded with a received ethernet frame, the ethernet data will begin at the start of the memory page boundary. Hence, the ethernet control information can be logically stripped from the end merely by adjusting the page's length field. By manipulating virtual memory mapping, this same page (sans ethernet control information), can then be passed to the local IP module without additional allocation and loading of memory. The disadvantage in using trailers is that it is non-standard. Many implementations cannot parse trailers.

The hanging FTP problem will appear if a gateway is not configured to recognize trailers, but a host or gateway immediately ''upstream'' on an ethernet uses them. Short datagrams will not be formatted with trailers, and so will be processed correctly. When the bulk data transfer starts, however, full-sized frames will be sent, and will use the trailer format. To the gateway that receives them, they appear simply as misformatted frames, and are quietly dropped. The solution, obviously, is to insure that all hosts and gateways on an ethernet are consistent in their use of trailers. Note that RFC 1122, ''Internet Host Requirements,'' places very strict restrictions on the use of trailers.

---

* To avoid this problem, TCP implementations on the Internet must use ''exponential backoff'' between successive retransmissions, Karn's algorithm for filtering samples used to estimate round-trip delay between TCP peers, and Jacobson's algorithm for incorporating variance into the ''retransmission time-out'' computation for TCP segments. See Section 4.2.3.1 of RFC 1122, ''Requirements for Internet Hosts — Communication Layers.''

4. Performance Testing

Performance management encompasses two rather different activities. One is passive system monitoring to detect problems and determine operational baselines. The goal is to measure system and component utilization and so locate bottlenecks, since bottlenecks should receive the focus of performance tuning efforts. Also, performance data is usually required by upper level management to justify the costs of communications systems. This is essentially identical to system monitoring, and is addressed at greater length in Section 2, above.

Another aspect of performance management is active performance testing and capacity planning. Some work in this area can be based on analysis. For example, a rough estimate of gateway capacity can be deduced from a simple model given by Charles Hedrick in his ''Introduction to Administration of an Internet-based Local Network,'' which is

```
per-packet processing time =
                switching time +
                                (packet size) * (transmission bps).
```

Another guideline for capacity planning is that in order to avoid excessive queuing delays, a system should be sized at about double its expected load. In other words, system capacity should be so high that utilization is no greater than 50%.

Although there are more sophisticated analytic models of communications systems than those above, their added complexity does not usually gain a corresponding accuracy. Most analytic models of communications nets require assumptions about traffic load distributions and service rates that are not merely problematic, but are patently false. These errors tend to result in underestimating queuing delays. Hence, it is often necessary to actually load and measure the performance of a real communications system if one is to get accurate performance predictions. Obviously, this type of testing is performed on isolated systems or during off hours. The results can be used to evaluate parameter settings or predict performance during normal operations.

Simulations can be used to supplement the testing of real systems. To be believable, however, simulations require validation, which, in turn, requires measurements from a real system. Whether testing or simulating a system's performance, actual traffic traces should be incorporated as input to traffic generators. The performance of a communications system will be greatly influenced by its load characteristics (burstiness, volume, etc.), which are themselves highly dependent on the applications that are run.

When tuning a net, in addition to the usual configuration parameters, consider the impact of the location of gateways and print and file servers. A few rules of thumb can guide the location of shared system resources. First, there is the principle of locality: a system will perform better if most traffic is between nearby destinations. The second rule is to avoid creating bottlenecks. For example, multiple diskservers may be called for to support a large number of workstations. Furthermore, to avoid LAN and diskserver congestion, workstations should be configured with enough memory to avoid frequent swapping.

As a final note on performance management, proceed cautiously if your ethernet interface allows you to customize its collision recovery algorithm. This is almost always a bad idea. The best that it can accomplish is to give a few favored hosts a disproportionate share of the ethernet bandwidth, perhaps at the cost of a reduction in total system throughput. Worse, it is possible that differing collision recovery algorithms may exhibit a self-synchronizing behavior, so that excess collisions are generated.

5. Configuration Management

Configuration management is the setting, collecting, and storing of the state and parameters of network resources. It overlaps all other network management functions. Hence, some aspects of configuration management have already been addressed (e.g., tuning for performance). In this section, we will focus on configuration management activities needed to ''hook up'' a net or campus to a larger internet. We will not, of course, include specific details on installing or maintaining internetted communications systems. We will, however, skim over some of the TCP/IP configuration highlights.

Configuration management includes ‘‘name management’’ — the control and allocation of system names and addresses, and the translation between names and addresses. Name-to-address translation is performed by ‘‘name servers.’’ We conclude this section with a few strictures on the simultaneous use of two automated name-servers, the Domain Name System (DNS), and Yellow Pages (YP).

5.1 Required Host Configuration Data for TCP/IP internets

In a TCP/IP internet, each host needs several items of information for internet communications. Some will be host-specific, while other information will be common for all hosts on a subnet. In a soon to be published RFC document,* R. Droms identifies the following configuration data required by internet hosts:

- An IP address, a host specific value that can be hard-coded or obtained via BOOTP, the Reverse Address Resolution Protocol (RARP) or Dynamic RARP (DRARP).
- Subnet properties, such as the subnet mask and the Maximum Transmission Unit (MTU); obviously, these values are not host-specific.
- Addresses of ‘‘entry’’ gateways to the internet; addresses of default gateways are usually hard-coded; though the ICMP ‘‘redirect’’ message can be used to refine a host’s routing tables, there is currently no dynamic TCP/IP mechanism or protocol for a host to locate a gateway; an IETF working group is busy on this problem.
- For hosts in internets using the Domain Name System (DNS) for name-to-address translation, the location of a local DNS server is needed; this information is not host-specific, and usually hard-coded;
- Host name (domain name, for hosts using DNS); obviously host-specific; either hard-coded or obtained in a boot procedure.
- For diskless hosts, various boot services. BOOTP is the standard Internet protocol for downloading boot configuration information. The Trivial File Transfer Protocol (TFTP) is typically used for downloading boot images. Sun computers use the ‘‘bootparams’’ RPC mechanism for downloading initial configuration data to a host.

There are ongoing developments, most notably the work of the Dynamic Host Configuration Working Group of the IETF, to support dynamic, automatic gathering of the above data. In the meantime, most systems will rely on hand-crafted configuration files.

5.2 Configuration Guidelines for Any TCP/IP internet

An IP address consists of a network identifier, an optional subnet identifier, and a host identifier. None of these fields can be assigned arbitrarily.

Internet routing is based on the network identifier. Separate, partitioned nets advertising the same IP network number will cause routing chaos. To avoid collisions in network identifiers, selection must be coordinated with the internet administration.

If used, the subnet identifier of an IP address must be at least two bits long. The subnet identifier usually occupies contiguous bits, though this is not necessary. When initially configuring a network, some thought should be given to how many subnets will eventually be needed. There is a natural tendency to underestimate the growth of a communications system.

On a single net, each host must have a unique host identifier. Another constraint on address assignment is that the use of all 0’s as the host number in an IP address is a bad idea. It may work in limited cases, but it is contrary to the specifications, and will fail if pushed.

---

* Draft ‘‘Dynamic Configuration of Internet Hosts.’’

Each host needs router information, at least to include a default gateway. Gateways need to be configured to use the appropriate route protocol (e.g., RIP, EGP). Gated is a flexible, though complex process that can simultaneously run RIP, Hello, and EGP, and soon will be able to run BGP and OSPF. Consult with Mark Fedor (fedor@patton.NYSER.NET) if you plan to run gated.

Special care should be taken to ensure that all system components share a common format for IP broadcast addresses. Historically, there have been several broadcast formats. Unfortunately, disaster, in the form of broadcast storms, can result if broadcast addresses are mistaken for specific destinations and are forwarded. For this and other reasons, a host with a single IP interface should NOT be configured to relay packets.

5.3 Connecting to THE Internet

The original TCP/IP Internet (spelled with an upper-case ‘‘I’’) is still active, and still growing. An interesting aspect of the Internet is that it spans many independently administered systems.

Connection to the Internet requires: a registered network number, for use in IP addresses; a registered autonomous system number (ASN), for use in internet routing; and, a registered domain name. Fielding a primary and backup DNS server is a condition for registering a domain name.

The Defense Data Network (DDN) Network Information Center (NIC) is responsible for registering network numbers, autonomous system numbers, and domain names. Regional nets will have their own policies and requirements for Internet connections, but all use the NIC for this registration service. Contact the NIC for further information, at:

DDN Network Information Center
SRI International, Room EJ291
333 Ravenswood Avenue
Menlo Park, CA 94025

Email:	HOSTMASTER@NIC.DDN.MIL
Phone:	1-415-859-3695
	1-800-235-3155 (toll-free hotline)

5.4 YP and DNS: Dueling name servers.

The Domain Name System (DNS) provides name service: it translates host names into IP addresses (this mapping is also called ‘‘resolution’’). Two widespread DNS implementations are ‘‘bind’’ and ‘‘named.’’ The Sun Yellow Pages (YP) system can be configured to provide an identical service, by providing remote, keyed access to the ‘‘hosts.byname’’ map. Unfortunately, if both DNS and the YP hosts.byname map are installed, they can interact in disruptive ways.

The problem has been noted in systems in which DNS is used as a fallback, to resolve hostnames that YP cannot. If DNS is slow in responding, the timeout in program ypserv may expire, which triggers a repeated request. This can result in disaster if DNS was initially slow because of congestion: the slower things get, the more requests are generated, which slows things even more. A symptom of this problem is that failures by the DNS server or network will trigger numerous requests to DNS.

Reportedly, the bug in YP that results in the avalanche of DNS requests has been repaired in SunOS 4.1. The problem, however, is more fundamental than an implementation error. The YP map hosts.byname and the DNS contain the same class of information. One can get an answer to the same query from each system. These answers may well be different: there is not a mechanism to maintain consistency between the systems. More critical, however, is the lack of a mechanism or procedure to establish which system is authoritative. Hence, running the DNS and YP name services in parallel is pointless. If the systems stay consistent, then only one is needed. If they differ, there is no way to choose which is correct.

The YP hosts.byname service and DNS are comparable, but incompatible. If possible, a site should not run both services. Because of Internet policy, sites with Internet connections MUST use the DNS. If YP is also used, then it should be configured with YP-to-DNS disabled.

Hacking a system so that it uses DNS rather than the YP hosts.byname map is not trivial, and should not be attempted by novices. The approach is to rebuild the shared C link-library, so that system calls to gethostbyname() and gethostbyaddr() will use DNS rather than YP. To complete the change, programs that do not dynamically link the shared C library (rcp, arp, etc.) must also be rebuilt.

Modified shared C libraries for Sun 3s and Sun 4s are available via anonymous FTP from host uunet.uu.net, in the sun-fixes directory. Note that use of DNS routines rather than YP for general name resolution is not a supported SunOS feature at this time.

6. Internet Security

The guidelines and advice in this section pertain to enhancing the protection of data that are merely ‘‘sensitive.’’ By themselves, these measures are insufficient for protecting ‘‘classified’’ data. Implementing the policies required to protect classified data is subject to stringent, formal review procedures, and is regulated by agencies such as the Defense Investigative Service (DIS) and the National Security Agency (NSA).

A network manager must realize that he is responsible for protecting his system and its users. Furthermore, though the Internet may appear to be a grand example of a cooperative joint enterprise, recent incidents have made it clear that not all Internet denizens are benign.

A network manager should be aware that the network services he runs have a large impact on the security risks to which his system is exposed. The prudent network manager will be very careful as to what services his site provides to the rest of the Internet, and what access restrictions are enforced. For example, the protocol ‘‘finger’’ may provide more information about a user than should be given to the world at large. Worse, most implementations of the protocol TFTP give access to all world-readable files.

This section highlights several basic security considerations for Internet sites. It then lists several sources of information and advice on improving the security of systems connected to the Internet.

6.1 Basic Internet Security

Two major Internet security threats are denial of service and unauthorized access.

Denial of service threats often take the form of protocol spoofers or other malicious traffic generators. These problems can be detected through system monitoring logs. If an attack is suspected, immediately contact your regional net office (e.g., SURANET, MILNET). In addition, DDN users should contact SCC, while other Internet users should contact CERT (see below). A cogent description of your system’s symptoms will be needed.

At your own site, be prepared to isolate the problems (e.g., by limiting disk space available to the message queue of a mail system under attack). As a last resort, coping with an attack may require taking down an Internet connection. It is better, however, not to be too quick to quarantine your site, since information for coping with the attack may come via the Internet.

Unauthorized access is a potentially more ominous security threat. The main avenues are attacks against passwords and attacks against privileged system processes.

An appallingly common means of gaining entry to systems is by use of the initial passwords to root, sysdiag, and other management accounts that systems are shipped with. Only slightly less vulnerable are common or trivial passwords, since these are readily subverted by dictionary attacks.* Obvious steps can reduce the risk of password attacks: passwords should be short-lived, at least eight characters long, with a mix of upper and lower case, and preferably random. The distasteful aspect of memorizing a random string can be alleviated if the

* Exotic fantasy creatures and women’s names are well represented in most password dictionaries.

password is pronounceable.

Improving passwords does not remove all risks. Passwords transmitted over an ethernet are visible to all attached systems. Furthermore, gateways have the potential to intercept passwords used by any FTP or TELNET connections that traverse them. It is a bad idea for the root account to be accessed by FTP or TELNET if the connections must cross untrusted elements.

Attacks against system processes are another avenue of unauthorized access. The principle is that by subverting a system process, the attacker can then gain its access privileges.

One approach to reducing this risk is to make system programs harder to subvert. For example, the widespread attack in November 1988 by a self-replicating computer program (''worm,'' analogous to a tapeworm) subverted the ''fingerd'' process, by loading an intrusive bootstrap program (known variously as a ''grappling hook'' or ''vector'' program), and then corrupting the stack space so that a subroutine's return address was overwritten with the address of the bootstrap program.** The security hole in fingerd consisted of an input routine that did not have a length check. Security fixes to ''fingerd'' include the use of a revised input routine.

A more general protection is to apply the principle of ''least privilege.'' Where possible, system routines should run under separate user IDs, and should have no more privilege than is necessary for them to function.

To further protect against attacks on system processes, system managers should regularly check their system programs to ensure that they have not been tampered with or modified in any way. Checksums should be used for this purpose. Using the operating system to check a file's last date of modification is insufficient, since the date itself can be compromised.

Finally, to avoid the unauthorized replacement of system code, care should be exercised in assigning protection to its directory paths.

Some system programs actually have ''trap doors'' that facilitate subversion. A trap door is the epitome of an undocumented feature: it is a hidden capability of a system program that allows a knowledgeable person to gain access to a system. The Internet Worm exploited what was essentially a trap door in the BSD sendmail program.

Ensuring against trap doors in software as complex as sendmail may be infeasible. In an ideal world, the BSD sendmail program would be replaced by an entire mail subsystem (i.e., perhaps including mail user agents, mail transfer agents, and text preparation and filing programs). Any site using sendmail should at least obtain the less vulnerable, toughened distribution from ucbarpa.berkeley.edu, in file ˜ftp/4.3/sendmail.tar.Z. Sites running SunOS should note that the 4.0.3 release closed the security holes exploited by the Internet Worm. Fixes for a more obscure security hole in SunOS are available from host uunet.uu.net in ˜ftp/sun-fixes; these improvements have been incorporated in SunOS 4.1.

Sendmail has problems other than size and complexity. Its use of root privileges, its approach to alias expansion, and several other design characteristics present potential avenues of attack. For UNIX sites, an alternative mail server to consider is MMDF, which is now at version 2. MMDF is distributed as part of the SCO UNIX distribution, and is also available in the user contributed portion of 4.3BSD. Though free, MMDF is licensed, and resale is restricted. Sites running MMDF should be on the mmdf email list; requests to join this list should be sent to:

mmdf2-request@relay.cs.net.

Programs that masquerade as legitimate system code but which contain trap doors or other aides to unauthorized access are known as trojan horses. Computer ''viruses,'' intrusive software that infects seemingly innocent programs and propagates when the infected programs are executed or copied, are a special case of trojan horse programs.*

** An early account of the Internet Worm incident of November 1988 is given by Eugene Spafford in the January 89 issue of ''Computer Communications Review.'' Several other articles on the worm incident are in the June 89 issue of the ''Communications of the ACM.''

* Virus attacks have been seen against PCs, but as yet have rarely been directed agains UNIX systems.

To guard against trojan horse attacks, be wary of programs downloaded from remote sources. At minimum, do not download executables from any but the most trusted sources. Also, as noted above, to avoid proliferation of ‘‘infected’’ software, checksums should be computed, recorded, and periodically verified.

6.2 Security Information Clearing-Houses

The Internet community can get security assistance from the Computer Emergency Response Team (CERT), established by DARPA in November 1988. The Coordination Center for the CERT (CERT/CC) is located at the Software Engineering Institute at Carnegie Mellon University. The CERT is intended to respond to computer security threats such as the November ’88 worm attack that invaded many defense and research computers. Consult RFC 1135 (Reynolds, J., "The Helminthiasis of the Internet", USC/ISI, December 1989), for further information.

CERT assists Internet sites in response to security attacks or other emergency situations. It can immediately tap experts to diagnose and solve the problems, as well as establish and maintain communications with the affected computer users and with government authorities as appropriate. Specific responses will be taken in accordance with the nature of the problem and the magnitude of the threat.

CERT is also an information clearing-house for the identification and repair of security vulnerabilities, informal assessments of existing systems in the research community, improvement to emergency response capability, and both vendor and user security awareness. This security information is distributed by periodic bulletins, and is posted to the USENET news group comp.security.announce. In addition, the security advisories issued by CERT, as well as other useful security-related information, are available via anonymous FTP from cert.sei.cmu.edu.

For immediate response to attacks or incidents, CERT mans a 24-hour hotline at (412) 268-7090. To subscribe to CERT’s security announcement bulletin, or for further information, contact:

CERT
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

(412) 268-7080
cert@cert.sei.cmu.edu.

For DDN users, the Security Coordination Center (SCC) serves a function similar to CERT. The SCC is the DDN’s clearing-house for host/user security problems and fixes, and works with the DDN Network Security Officer. The SCC also distributes the DDN Security Bulletin, which communicates information on network and host security exposures, fixes, and concerns to security and management personnel at DDN facilities. It is available online, via kermit or anonymous FTP, from nic.ddn.mil, in SCC:DDN-SECURITY-yy-nn.TXT (where ‘‘yy’’ is the year and ‘‘nn’’ is the bulletin number). The SCC provides immediate assistance with DDN-related host security problems; call (800) 235-3155 (6:00 a.m. to 5:00 p.m. Pacific Time) or send e-Mail to SCC@NIC.DDN.MIL. For 24 hour coverage, call the MILNET Trouble Desk (800) 451-7413 or AUTOVON 231-1713.

The CERT/CC and the SCC communicate on a regular basis and support each other when problems occur. These two organizations are examples of the incident response centers that are forming; each serving their own constituency or focusing on a particular area of technology.

Other network groups that discuss security issues are: comp.protocols.tcp-ip, comp.virus (mostly PC-related, but occasionally covers Internet topics), misc.security, and the BITNET Listserv list called VIRUS-L.

7. Internet Information

There are many available references on the TCP/IP protocol suite, the internet architecture, and the DDN Internet. A soon to be published FYI RFC document, ‘‘Where to Start: A Bibliography of General Internetworking Information.’’ provides a bibliography of online and hard copy documents, reference materials, and multimedia training tools that address general networking information and ‘‘how to use the Internet.’’ It presents a representative collection of materials that will help the reader become familiar with the concepts of internetworking. Inquires on the current status of this document can sent to user-doc@nnsc.nsf.net or by postal mail to:

Corporation for National Research Initiatives
1895 Preston White, Suite 100
Reston, VA 22091
Attn: IAB Secretariat.

Two texts on networking are especially noteworthy. *Internetworking With TCP/IP*, by Douglas Comer, is an informative description of the TCP/IP protocol suite and its underlying architecture. The *UNIX System Administration Handbook,* by Nemeth, Snyder, and Seebass, is a ‘‘must have’’ for system administrators who are responsible for UNIX hosts. In addition to covering UNIX, it provides a wealth of tutorial material on networking, the Internet, and network management.

A great deal of information on the Internet is available online. An automated, online reference service is available from CSNET. To obtain a bibliography of their online offerings, send the email message

request: info
topic: help
request: end

to info-server@sh.cs.net.

The DDN NIC also offers automated access to many NIC documents, online files, and WHOIS information via electronic mail. To use the service, send an email message with your request specified in the SUBJECT field of the message. For a sampling of the type of offerings available through this service, send the following message

To: SERVICE@NIC.DDN.MIL
Subject: help
Msg: <none>

The DDN Protocol Implementations and Vendors Guide, published by the DDN Network Information Center (DDN NIC),* is an online reference to products and implementations associated with the DoD Defense Data Network (DDN) group of communication protocols, with emphasis on TCP/IP and OSI protocols. It contains information on protocol policy and evaluation procedures, a discussion of software and hardware implementations, and analysis tools with a focus on protocol and network analyzers. To obtain the guide, invoke FTP at your local host and connect to host NIC.DDN.MIL (internet address 26.0.0.73 or 10.0.0.51). Log in using username ’anonymous’ with password ’guest’ and get the file NETINFO:VENDORS-GUIDE.DOC.

The DDN Protocol Guide is also available in hardcopy form. To obtain a hardcopy version of the guide, contact the DDN Network Information Center:

By U.S. mail:
SRI International
DDN Network Information Center

* Products mentioned in the guide are not specifically endorsed or recommended by the Defense Communications Agency (DCA).

333 Ravenswood Avenue, Room EJ291
Menlo Park, CA 94025

By e-mail:
NIC@NIC.DDN.MIL

By phone:
1-415-859-3695
1-800-235-3155 (toll-free hotline)

For further information about the guide, or for information on how to list a product in a subsequent edition of the guide, contact the DDN NIC.

There are many additional online sources on Internet Management. RFC 1118, ‘‘A Hitchhiker’s Guide to the Internet,’’ by Ed Krol, is a useful introduction to the Internet routing algorithms. For more of the nitty-gritty on laying out and configuring a campus net, see Charles Hedrick’s ‘‘Introduction to Administration of an Internet-based Local Network,’’ available via anonymous FTP from cs.rutgers.edu (sometimes listed in host tables as aramis.rutgers.edu), in subdirectory runet, file tcp-ip-admin. Finally, anyone responsible for systems connected to the Internet must be thoroughly versed in the Host Requirements RFCs (RFC 1122 and RFC 1123) and ‘‘Requirements for Internet Gateways,’’ RFC 1009.

8. The Final Words on Internet Management

Keep smiling, no matter how bad things may seem. You are the expert. They need you.

9. Security Considerations

Security issues are discussed in Section 6.

10. Author’s Address

Robert H. Stine
SPARTA, Inc.
7926 Jones Branch Drive
Suite 1070
McLean, VA 22102

EMail: STINE@SPARTA.COM