

Network Management

Lecture 10

Network Management Tools, Systems, and Engineering

Basic Network Software Tools

- Status monitoring tools
- Traffic monitoring tools
- Route monitoring tools

Notes

- Basic tools are available as
 - Part of the Operating System
 - Add-on applications

Status Monitoring Tools

Table 9.1 Status-Monitoring Tools

NAME	OS	DESCRIPTION
ifconfig	UNIX	Obtains and configures networking interface parameters and status
ping	UNIX Windows	Checks the status of node / host
nslookup	UNIX Windows	Looks up DNS for name-IP address translation
dig	UNIX	Queries DNS server
host	UNIX	Displays information on Internet hosts / domains

Notes

ifConfig

- Used to assign/read an address to/of an interface
- Option -a is to display all interfaces
- Notice two interface loop-back (lo0) and Ethernet (hme0)

Notes

Example:

```
netman: ifconfig -a
```

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232  
    inet 127.0.0.1 netmask ff000000
```

```
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,  
    MULTICAST> mtu 1500 inet 192.207.8.31 netmask fffff00  
    broadcast 192.207.8.255
```

Ping

- Most basic tool for internet management
- Based on ICMP ECHO_REQUEST message
- Available on all TCP/IP stacks
- Useful for measuring connectivity
- Useful for measuring packet loss
- Can do autodiscovery of TCP/IP equipped stations on single segment

Notes

Example:

```
% ping 205.152.8.138
PING 205.152.8.138 (205.152.8.138): 56 data bytes
64 bytes from 205.152.8.138: icmp_seq=0 ttl=17 time=14.8 ms
64 bytes from 205.152.8.138: icmp_seq=1 ttl=17 time=20.2 ms
64 bytes from 205.152.8.138: icmp_seq=2 ttl=17 time=15.7 ms
64 bytes from 205.152.8.138: icmp_seq=3 ttl=17 time=21.6 ms
64 bytes from 205.152.8.138: icmp_seq=4 ttl=17 time=20.0 ms
```

```
--- 205.152.8.138 ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 14.8/18.4/21.6 ms
```

nslookup

- An interactive program for querying Internet Domain Name System servers
- Converts a hostname into an IP address and vice versa querying DNS
- Useful to identify the subnet a host or node belongs to
- Lists contents of a domain, displaying DNS record
- Available with BSD UNIX; FTP from uunet.uu.net
- Available in Windows NT

Notes

Example:

```
noc2% nslookup 172.152.8.138
```

```
Server: ada.btc.gatech.edu
```

```
Address: 192.77.147.28
```

```
Name: mani.bellsouth.net
```

```
Address: 172.152.8.138
```

Domain Name Groper: dig

Used to gather lots of information on hosts
from DNS

Notes

Example:

```
[beluga]~> dig +nocomments nimbus.tenet.res.in
nimbus.tenet.res.in. 604800  IN      A      203.199.255.4
tenet.res.in.        604800  IN      NS     volcano.tenet.res.in.
tenet.res.in.        604800  IN      NS     lantana.tenet.res.in.
volcano.tenet.res.in.604800  IN      A      203.199.255.3
;; Query time: 2 msec
;; SERVER: 203.199.255.3#53(203.199.255.3)
;; WHEN: Fri Mar 6 14:12:43 2009
;; MSG SIZE rcvd: 149
[beluga]~>
```

Host

- Command: host
- Displays host names using DNS
- Available from ftp.nikhef.nl:/pub/network/host.tar.Z

Notes

Example:

```
% host -a sun4-gw.cc.gatech.edu
```

```
Trying null domain
```

```
rcode = 0 (Success), ancourt=1
```

```
The following answer is not authoritative:
```

```
sun4-gw.cc.gatech.edu 85851 IN A 130.207.111.100
```

Traffic Monitoring Tools

Table 9.2 Traffic-Monitoring Tools

Name	Operating System	Description
ping	UNIX / Windows	Used for measuring roundtrip packet loss
bing	UNIX	Measures point-to-point bandwidth of a link
tcpdump	UNIX	Dumps traffic on a network
getethers	UNIX	Acquires all host addresses of an Ethernet LAN segment
iptrace	UNIX	Measures performance of gateways
ethereal, wireshark	Linux / Windows	Graphical tool to capture, inspect , and to save Ethernet packets

Notes

- *ping* and *bing* used to measure the propagation characteristics of the transmission path
- *ethereal* (a.k.a. *wireshark*), and *tcpdump* (also *snoop*) puts the network interface in promiscuous mode and logs the data
- *iptrace* uses NETMON program in UNIX and produces 3 types of outputs:
 - IP traffic
 - Host traffic matrix
 - Abbreviated sampling of pre-defined number of packets

Packet Loss Measurement

- Command: ping
- Many options available
- Implementation varies from system to system

Notes

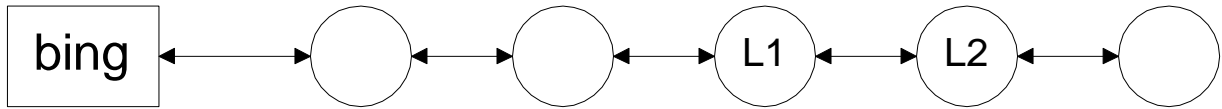
Example:

```
netman: ping -s mit.edu
PING mit.edu: 56 data bytes
64 bytes from MIT.MIT.EDU (18.72.0.100): icmp_seq=0. time=42. ms
64 bytes from MIT.MIT.EDU (18.72.0.100): icmp_seq=1. time=41. ms
64 bytes from MIT.MIT.EDU (18.72.0.100): icmp_seq=2. time=41. ms
64 bytes from MIT.MIT.EDU (18.72.0.100): icmp_seq=3. time=40. ms
64 bytes from MIT.MIT.EDU (18.72.0.100): icmp_seq=4. time=40. ms
```

```
----mit.edu PING Statistics----
```

```
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 40/40/42
```

bing



- Used to determine throughput of a link
- Uses `icmp_echo` utility
- Knowing packet size and delay, calculates bandwidth
- `bing L1` and `L2` and the difference yields the bandwidth of link L1-L2
- Bandwidth of link L1-L2 could be higher than the intermediate links.

Notes

Ethereal (Wireshark)

The screenshot shows the Wireshark interface with a packet capture list and details pane. The packet list shows various protocols including DHCP, ARP, and TCP. The details pane shows the structure of a selected packet, including Ethernet II, Internet Protocol, and Transmission Control Protocol.

No.	Time	Source	Destination	Protocol	Info
15751	13.599814	10.94.12.156	255.255.255.255	DHCP	DHCP Inform - Transaction ID 0xa62b9a27
15752	13.599997	10.94.1.1	255.255.255.255	DHCP	DHCP ACK - Transaction ID 0xa62b9a27
15753	13.600216	Intel_93:05:cf	Broadcast	ARP	Who has 10.94.86.242? Tell 10.94.7.159
15754	13.602520	QuantaCo_bb:96:6d	Broadcast	ARP	Who has 10.94.71.187? Tell 10.94.14.107
15755	13.603665	00000001.000fb0cd6cbf	00000000.ffffffffffff	IPX SAP	Nearest Query
15756	13.604452	Wistron_9c:2b:15	Broadcast	ARP	Who has 10.94.3.229? Tell 10.94.18.116
15757	13.604575	IntelCor_28:f7:54	Broadcast	ARP	Who has 10.94.196.215? Tell 10.94.13.79
15758	13.605370	Intel_37:51:d6	Broadcast	ARP	Who has 10.94.25.45? Tell 10.94.19.149
15759	13.606849	Intel_80:ee:93	Broadcast	ARP	Who has 10.94.48.201? Tell 10.94.8.254
15760	13.607949	10.6.21.59	10.94.3.215	TCP	48966 > http [SYN] Seq=0 Win=5840 Len=0 MSS=1460 TSV=115648 TSER=0 WS=5
15761	13.607974	10.94.3.215	10.6.21.59	TCP	http > 48966 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 WS=6
15762	13.607983	Dell_f2:16:e3	Broadcast	ARP	Who has 10.94.22.27? Tell 10.94.13.242
15763	13.608379	10.6.21.59	10.94.3.215	TCP	48966 > http [ACK] Seq=1 Ack=1 Win=5856 Len=0
15764	13.608624	AsustekC_92:53:5a	Broadcast	ARP	Who has 10.94.204.43? Tell 10.94.33.110
15765	13.610549	Giga-Byt_ce:d1:c3	Broadcast	ARP	Who has 10.94.70.165? Tell 10.94.35.9
15766	13.611760	AsustekC_ef:87:30	Broadcast	ARP	Who has 10.94.81.89? Tell 10.94.19.251

Details pane for selected packet:

- Frame 1 (60 bytes on wire, 60 bytes captured)
- Ethernet II, Src: Cisco_df:64:40 (00:11:bc:df:64:40), Dst: Intel_9c:c6:a2 (00:19:d1:9c:c6:a2)
- Internet Protocol, Src: 10.6.21.112 (10.6.21.112), Dst: 10.94.3.215 (10.94.3.215)
- Transmission Control Protocol, Src Port: 40261 (40261), Dst Port: ssh (22), Seq: 1, Ack: 1, Len: 0

Hex dump:

```

0000 00 19 d1 9c c6 a2 00 11 bc df 64 40 08 00 45 10  ....d@..E.
0010 00 28 c8 f9 40 00 3e 06 46 1c 0a 06 15 70 0a 5e  .(..@.>.F...p.^
0020 03 d7 9d 45 00 16 57 49 28 09 71 cd 7e c1 50 10  ...E..WI (.q.~.P.
0030 4d 0a 27 e3 00 00 00 00 00 00 00 00  ....M.'.....

```

File: "/tmp/etherXXXXUNr6d6" 3222 K... ; Packets: 22608 Displayed: 22608 Marked: 0 Dropped: 364

Notes

snoop

- Puts a network interface in promiscuous mode
 - Logs data on
 - Protocol type
 - Length
 - Source address
 - Destination address
 - Reading of user data limited to superuser
-

Notes

Example: Options: -d for device interface and
-c for counts

```
root@noc2:~# snoop -d hme0 -c 5
Using device /dev/hme (promiscuous mode)
noc2.btc.gatech.edu -> noc4.btc.gatech.edu TCP D=22 S=1221
Ack=2845521735 Seq=24552727 Len=0 Win=7368
? -> (multicast) ETHER Type=809B (EtherTalk (AppleTalk over
Ethernet)), size = 80 bytes
? -> (multicast) ETHER Type=809B (EtherTalk (AppleTalk over
Ethernet)), size = 86 bytes
noc2.btc.gatech.edu -> 199.77.147.255 UDP D=137 S=137 LEN=108
noc2.btc.gatech.edu -> 199.77.147.255 UDP D=137 S=137 LEN=108
noc2.btc.gatech.edu -> 199.77.147.255 UDP D=137 S=137 LEN=108
noc2.btc.gatech.edu -> 199.77.147.255 UDP D=137 S=137 LEN=108
? -> (broadcast) ETHER Type=8137 (Novell (old) NetWare IPX), size =
88 noc4.btc.gatech.edu -> noc2.btc.gatech.edu TCP D=1221 S=22
Ack=24552727 Seq=2845521735 Len=64 Win=8760
noc2.btc.gatech.edu -> noc4.btc.gatech.edu TCP D=22 S=1221
Ack=2845521799 Seq=24552727 Len=0 Win=7304
noc4.btc.gatech.edu -> noc2.btc.gatech.edu TCP D=1221 S=22
Ack=24552727 Seq=2845521799 Len=56 Win=8760
snoop: 5 packets captured
```

tcpdump

- Command: tcpdump
- Interprets and prints headers for:

Ethernet	IP	ICMP
TCP	UDP	NFS
ND	ARP	Appletalk
- Useful for examining and evaluating the TCP based traffic
- Available in UNIX system; FTP from [ftp.ee.lbl.gov](ftp://ftp.ee.lbl.gov)

Notes

Example: SNMP message

```
14:03:36.798269 noc1.btc.gatech.edu.snmp > noc3.btc.gatech.edu.164:  
Community = public  
GetResponse(196)  
Request ID = 4  
system.sysDescr.0 = "SunOS noc1 5.5.1 Generic_103640-08 sun4u"  
system.sysObjectID.0 = E:hp.2.3.10.1.2  
system.sysUpTime.0 = 247396453  
system.sysContact.0 = "Brandon Rhodes"  
system.sysName.0 = "noc1"  
system.sysLocation.0 = "BTC NM Lab"  
system.sysServices.0 = 72
```

Figure 5.17(b) Get-Response Message from Agent-to-Manager (After)

Network Routing Tools

Table 9.3 Route-Monitoring Tools

Name	Operating System	Description
netstat	UNIX	Displays the contents of various network-related data structures
arp rarp	UNIX, Windows 9x/00/NT	Displays and modifies the Internet-to-Ethernet address translation tables
tracert tracert	UNIX Windows	Traces route to a destination with routing delays

Notes

Network Status

netstat -r
Routing tables

Internet:

Destination	Gateway	Flags	Refs	Use	Netif	Expire
Default	gw.litech.net	UGC	44	541550	de0	
172.16.15.1	gw.litech.net	UGH	0	0	de0	
ah.litech.net	0:80:48:ee:74:b4	UHLW	9	2653683	de0	202
uucp.litech.net	uucp.litech.net	UH	0	0	lo0	
sip-17.litech.net	big	UH	0	5551	ppp3	
dip-244.litech.net	gw.litech.net	UGH	0	2472	de0	
univers-litech-gw	gw.litech.net	UGH	0	47	de0	
194.44.232	gw.isr.lviv.ua	Ugc	0	171831	ppp9	
OSPF-ALL.MCAST.NET	localhost	UH	1	86491	lo0	
OSPF-DSIG.MCAST.NE	localhost	UH	1	25127	lo0	

Figure 9.5 Routing Table using netstat -r

Notes

Route Tracing

- Command: traceroute (UNIX) / tracert (MS TIME-EXCEED error report)
- Available in most UNIX OS
- Windows)
- ICMP Also available from uc.msc.unm.edu
- Discovers route taken by packets from source to destination
- Useful for diagnosing route failures
- Useful for detecting bottleneck nodes

Notes

Trace Route Sample 1

Tracing route to mani.btc.gatech.edu [199.77.147.96]
over a maximum of 30 hops:

```
 1  2 ms  3 ms  3 ms  bims008001.bims.bellsouth.net [205.152.8.1]
 2  4 ms  2 ms  3 ms  172.16.11.2
 3  5 ms  4 ms  3 ms  172.16.4.2
 4  5 ms  3 ms  3 ms  bims011033.bims.bellsouth.net [205.152.11.33]
 5  4 ms  4 ms  4 ms  205.152.13.98
 6  *      *      *      Request timed out
 7  5 ms  9 ms 12 ms  205.152.2.249
 8 33 ms 31 ms 31 ms  Hssi0-0-0.GW2.ATL1.ALTER.NET [157.130.65.229]
 9 68 ms 10 ms 11 ms  105.ATM3-0-0.XR1.ATL1.ALTER.NET [146.188.232.66]
10 11 ms 14 ms 12 ms  195.ATM12-0-0.BR1.ATL1.ALTER.NET [146.188.232.49]
11 16 ms 14 ms 14 ms  atlanta1-br1.bbnplanet.net [4.0.2.141]
12 19 ms 15 ms 17 ms  atlanta2-br2.bbnplanet.net [4.0.2.158]
13 21 ms 56 ms 328 ms atlanta2-cr99.bbnplanet.net [4.0.2.91]
14 17 ms 18 ms 17 ms  192.221.26.3
15 32 ms 20 ms 18 ms  130.207.251.3
16 20 ms 17 ms 17 ms  mani.btc.gatech.edu [199.77.147.96]
```

Trace complete

Notes

Trace Route Sample 2

Tracing route to mani.btc.gatech.edu [199.77.147.96]
over a maximum of 30 hops:

1	3 ms	3 ms	4 ms	bims008001.bims.bellsouth.net [205.152.8.1]
2	3 ms	3 ms	2 ms	172.16.11.2
3	5 ms	4 ms	4 ms	172.16.4.2
4	5 ms	3 ms	4 ms	bims011033.bims.bellsouth.net [205.152.11.33]
5	7 ms	4 ms	4 ms	205.152.13.98
6	*	*	*	Request timed out
7	9 ms	8 ms	9 ms	205.152.2.249
8	228 ms	214 ms	191 ms	206.80.168.9
9	230 ms	246 ms	234 ms	maeeast.bbnplanet.net [192.41.177.2]
10	243 ms	222 ms	212 ms	vienna1-nbr2.bbnplanet.net [4.0.1.93]
11	230 ms	213 ms	202 ms	vienna1-nbr3.bbnplanet.net [4.0.5.46]
12	247 ms	227 ms	236 ms	vienna1-br2.bbnplanet.net [4.0.3.149]
13	228 ms	235 ms	238 ms	atlanta1-br1.bbnplanet.net [4.0.2.58]
14	*	257 ms	238 ms	atlanta2-br2.bbnplanet.net [4.0.2.158]
15	225 ms	234 ms	233 ms	atlanta2-cr99.bbnplanet.net [4.0.2.91]
16	240 ms	229 ms	251 ms	192.221.26.3
17	235 ms	245 ms	225 ms	130.207.251.3
18	*	268 ms	243 ms	mani.btc.gatech.edu [199.77.147.96]

Trace complete

Notes

SNMP Tools

- SNMP command-line tools
- SNMP MIB Browser with graphical interface
- snmpsniff: Linux/Free BSD based tool.
Reads PDUs

Notes

- Many tools available on public domain.

SNMP Command Tools

- snmptest
- snmpget
- snmpgetnext
- snmpset
- snmptrap
- snmpwalk
- snmpnetstat

Notes

- Test tool is an interactive tool to get values of several managed objects, one at a time.
- Get, Get-next and Set are the SNMP commands that we learned under SNMP architecture / messages. Execution of these will return an SNMP Response message.
- SNMPWalk uses snmpgetnext to trace the entire MIB.
- Network status command is used to test the status of network connections of a host.

SNMP Get Command

```
% snmpget noc5.btc.gatech.edu public  
system.sysDescr.0
```

```
system.sysDescr.0 = OCTET STRING: "SunOS  
noc5 5.6 Generic_105181-03 sun4u"
```

Notes

- Note that the value 0 at the end of the object id indicates that it is a single-valued scalar.

SNMP Get Next Command

```
% snmpgetnext noc5.btc.gatech.edu public  
interfaces.ifTable.ifEntry.ifIndex.1
```

```
interfaces.ifTable.ifEntry.ifIndex. 2 = INTEGER: 2
```

Notes

SNMP Set Command

- Command: `snmpset host community`

Notes

Network Status

- Command: `snmpnetstat host community`
- Useful for finding status of network connections

```
% snmpnetstat noc5 public
```

```
Active Internet Connections
```

```
Proto Recv-Q Send-Q Local Address Foreign Address (state)
tcp 0 0 *.* *.* CLOSED
tcp 0 0 localhost.46626 localhost.3456 ESTABLISHED
tcp 0 0 localhost.46626 localhost.3712 ESTABLISHED
tcp 0 0 localhost.46626 localhost.3968 ESTABLISHED
tcp 0 0 localhost.46626 localhost.4224 ESTABLISHED
tcp 0 0 localhost.3456 localhost.46626 ESTABLISHED
tcp 0 0 localhost.3712 localhost.46626 ESTABLISHED
tcp 0 0 localhost.3968 localhost.46626 ESTABLISHED
tcp 0 0 localhost.4224 localhost.46626 ESTABLISHED
tcp 0 0 noc5.41472 noc5.4480 ESTABLISHED
tcp 0 0 noc5.41472 noc5.4736 ESTABLISHED
tcp 0 0 noc5.4480 noc5.41472 ESTABLISHED
tcp 0 0 noc5.4736 noc5.41472 ESTABLISHED
```

Notes

SNMP Browser

- Command: `snmpwalk host community [variable name]`
- Uses Get Next Command
- Presents MIB Tree

Notes

199.77.147.182:

sysDescr.0 : SunOS noc5 5.6 Generic_105181-03 sun4u

sysObjectID.0 : 1.3.6.1.4.1.11.2.3.10.1.2

sysUpTime.0 : 8d 22:21:53.74

sysContact.0 :

sysName.0 : noc5

sysLocation.0 :

sysServices.0 : 72

sysORLastChange.0 : 0d 0:00:00.00

Figure 9.8 MIB Browser Example (text based) for System Group

SNMP Sniff

- *snmpsniff -I interface*
- A tool in Linux / FreeBSD environment
- Puts the interface in promiscuous mode and captures snmp PDUs.
- Similar to *tcpdump*

Notes

Protocol Analyzer

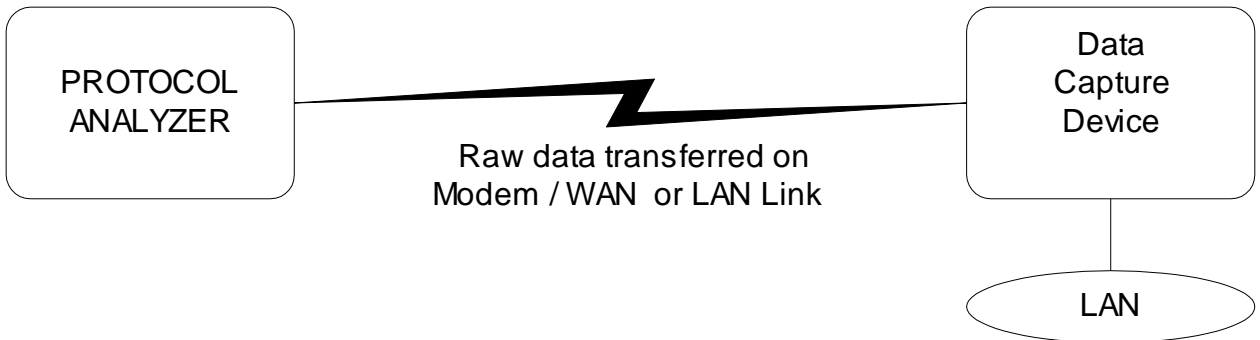


Figure 9.9 Basic Configuration of a Protocol Analyzer

Notes

- Analyzes data packets on any transmission line including LAN
- Measurements made locally or remotely
- Probe (data capture device) captures data and transfers to the protocol analyzer (no storage)
- Data link between probe and protocol analyzer either dial-up or dedicated link or LAN
- Protocol analyzer analyzes data at all protocol levels

RMON Probe

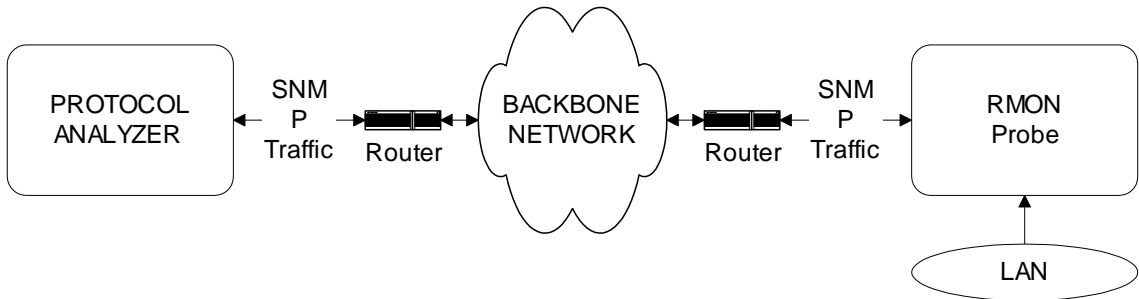


Figure 9.10 Protocol Analyzer with RMON Probe

Notes

- Network Associates Sniffer
 - Stand-alone and Networked
- HP NetMetrix / HP OpenView
 - Communication between probe and analyzer is using SNMP
- Data gathered and stored for an extended period of time and analyzed later
- Used for gathering traffic statistics and used for configuration management for performance tuning

Network Monitoring with RMON Probe

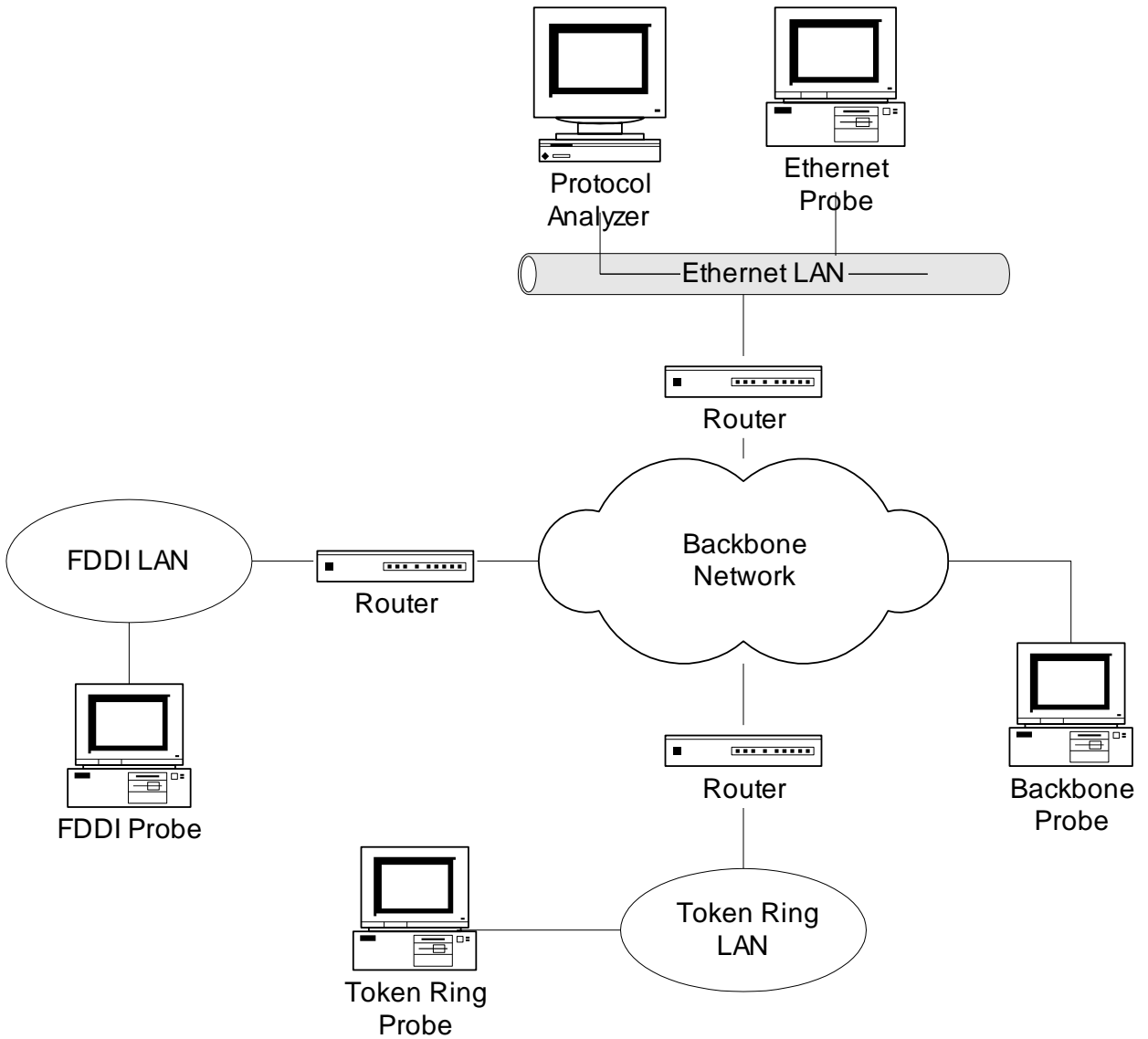


Figure 9.11 Monitoring of Total Network with Individual RMON Probes

Network Statistics

- Protocol Analyzers
- RMON Probe / Protocol analyzer
- MRTG (Multi router traffic grouper)
- Home-grown program using *tcpdump*

Notes

Traffic Load: Source

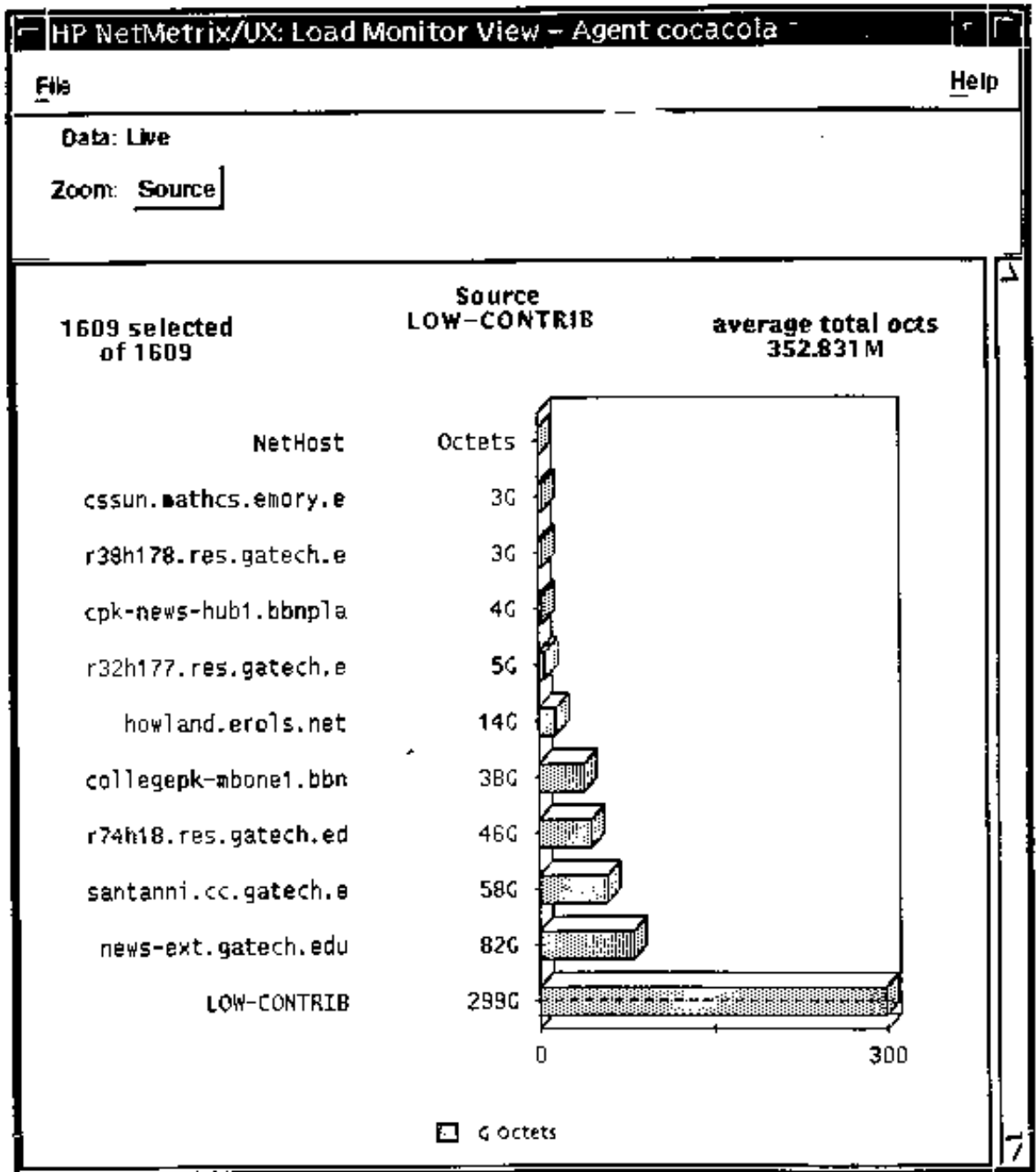


Figure 9.12 Load Statistics: Monitoring of Sources

Traffic Load: Destination

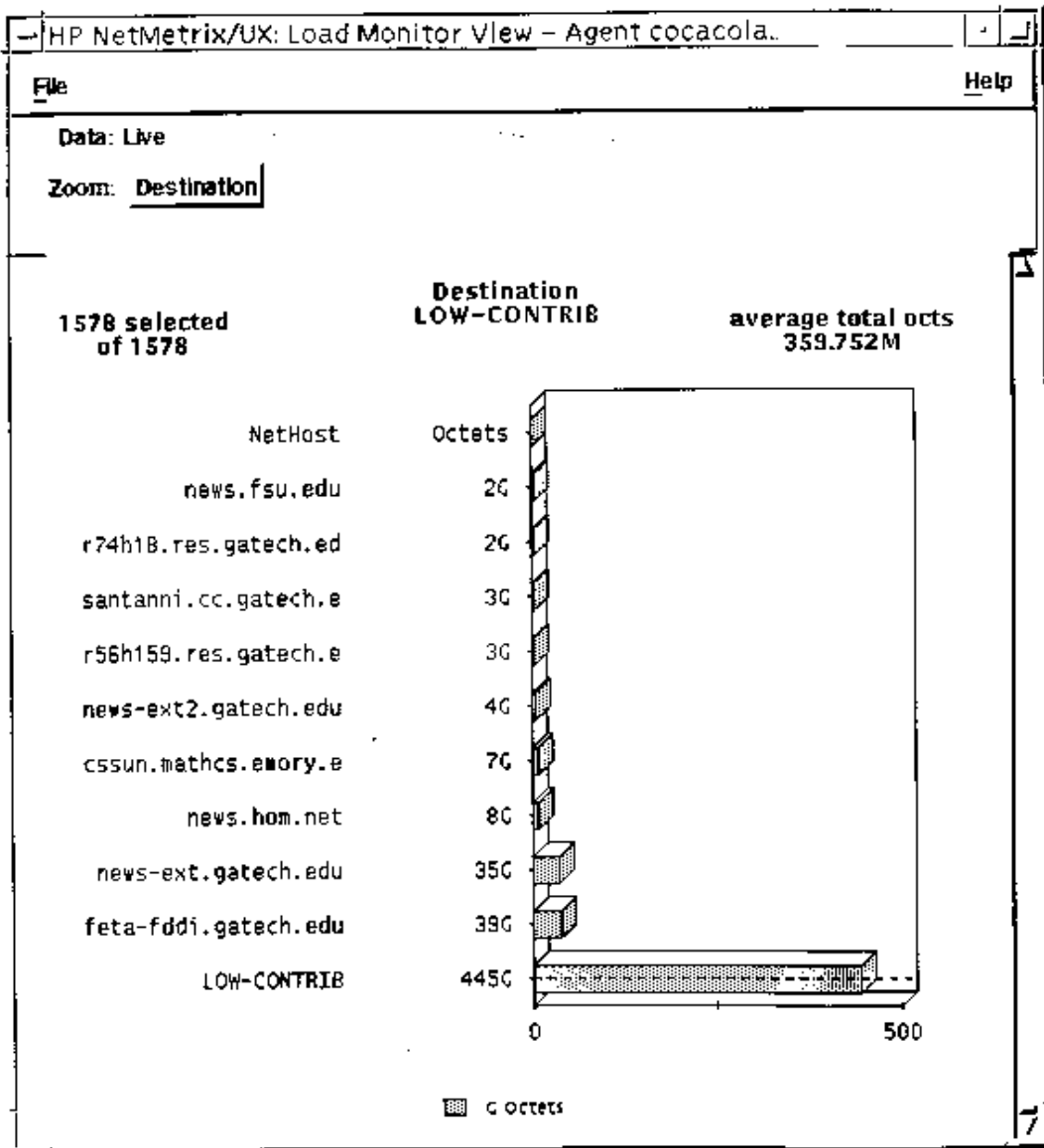


Figure 9.13 Load Statistics: Monitoring of Destinations

Traffic Load: Conversation

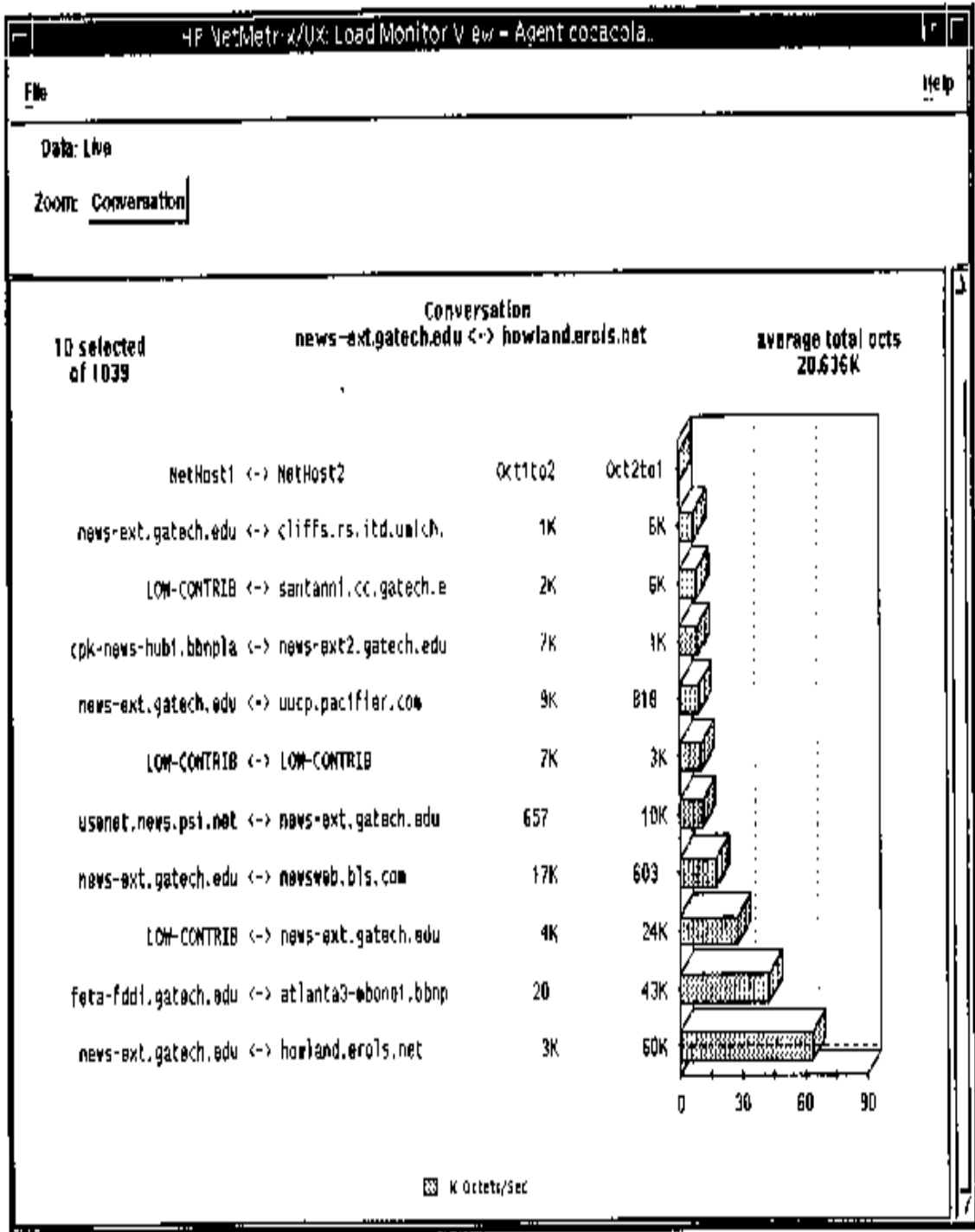


Figure 9.14 Load Statistics: Monitoring of Conversation Pairs

Protocol Distribution

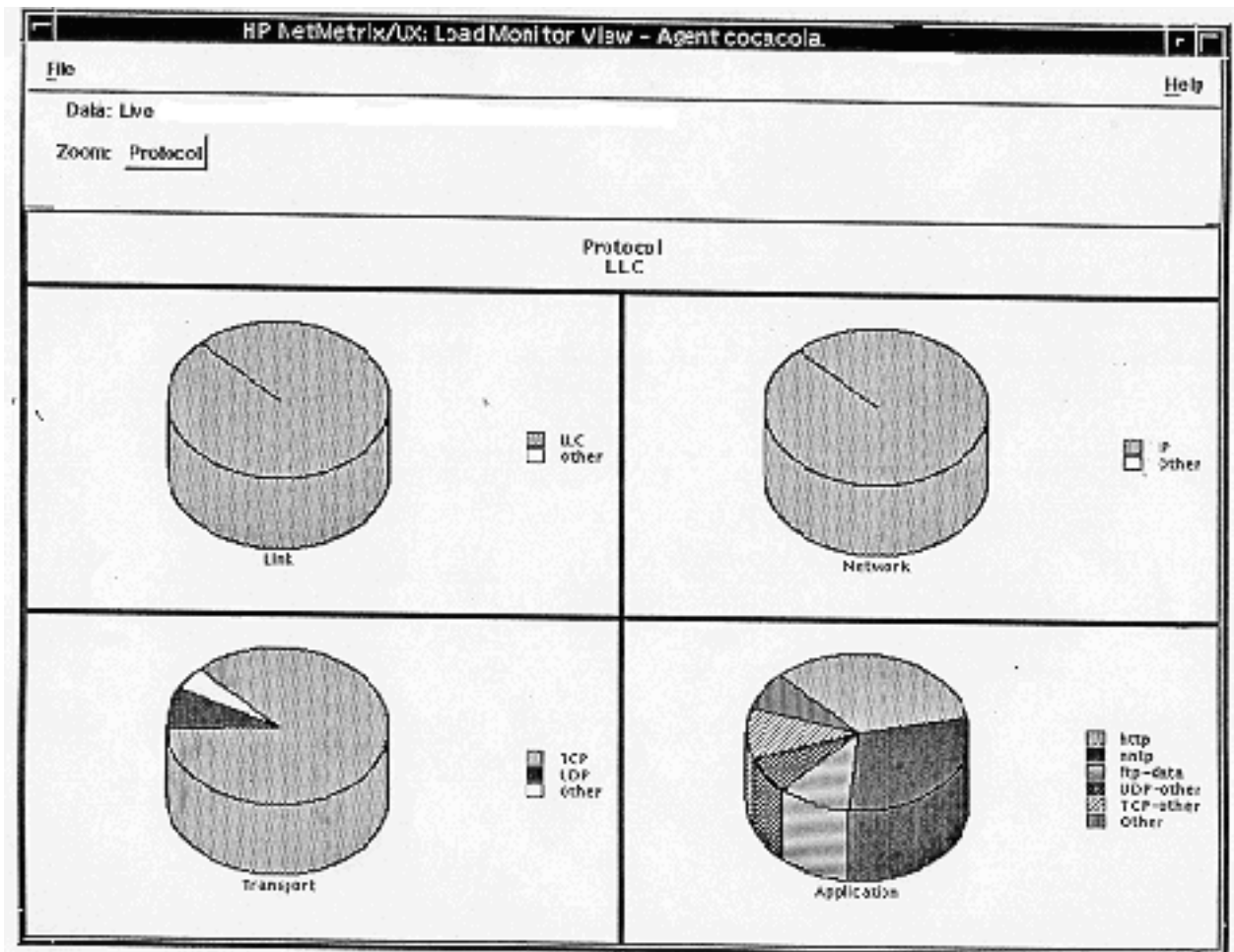


Figure 9.15 Protocol Distribution (NetMetrix)

MRTG

- Multi Router Traffic Grouper (Oeticker and Rand)
- www.ee.ethz.ch/stats/mrtg/
- Generates graphic presentation of traffic on Web
 - Daily view
 - Weekly view
 - Monthly view
 - Yearly view

Notes

MIB Engineering

MIB Engineering: What & Why

- Designing of MIB is **MIB Engineering**
- MIB contains SNMP model info of managed object
- Manager's view of the NE is limited to MIB view
- MIB view is embedded in NE and hence permanent
- Common constructs, **idioms**, used in MIB are defined by SMI

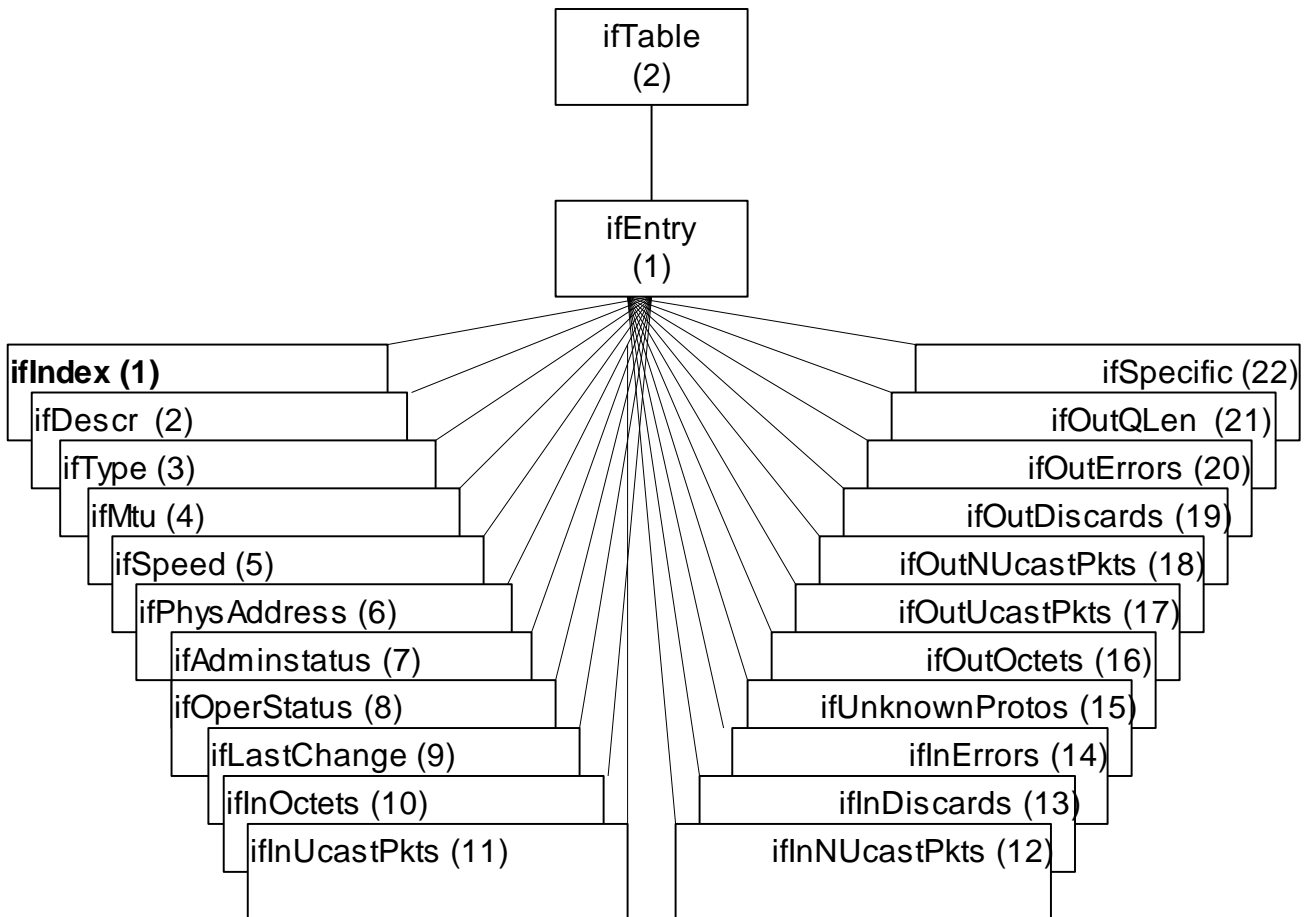
Notes

SMI: Principles

- Element defined by a set of variables, called objects
- Objects constricted to a limited number of data types
- Data types:
 - scalar, primitive or atomic type
e.g.: Boolean, Integer, IP address, String, counter,.
 - Constructed types :
 - Array:
 - Ordered set of elements
 - All elements are of same type
 - Each element identified by an index
 - Record:
 - Ordered collection of elements
 - Elements are of different types
 - Each element referred to by name
 - Set
 - Unordered collection of elements
 - Elements of same or different types

Notes

Example of Set: *ifTable*



Notes

- *ifTable* is an array of instantiation of *ifEntry*
- One element for each interface in the node with an *ifIndex*
- Each element in the table contains a record with several fields (variables) describing the interface

Example of Set: Trap Filter

- Definition of a trap filter in an agent

TrapFilter ::= SET OF Conditions

- Set of conditions are AND'ed together
- Trap forwarded when all conditions are satisfied
- Order of evaluation of the conditions irrelevant

Notes

SNMP / SMI Limitations

- **Restrictions on the construction of types**
 - Array can contain record and vice versa
 - Array can contain record that can contain other records
 - Array cannot contain a record that contains an array
 - Two records with the collection of identical records in two parts of a MIB has to have different names
- **SNMP limitation**
 - SNMP has variables and access methods, not methods to perform actions
- **Other restrictions include**
 - PDU size limits the number of objects for access
 - SNMP transaction very limited compared to db transactions

Notes

Counters vs Rates in NMS

- Rates play key role in performance measurements
- Rates impact of design of buffers in NEs
- Counters are used to measure rates
- Rate calculation depends on averaging time
- Different applications require rates based on different averaging time
- MIB does not include rates; NMS computes it
- Above factors should be considered in the NMS design

Notes

Object-Oriented Approach to MIB Engineering

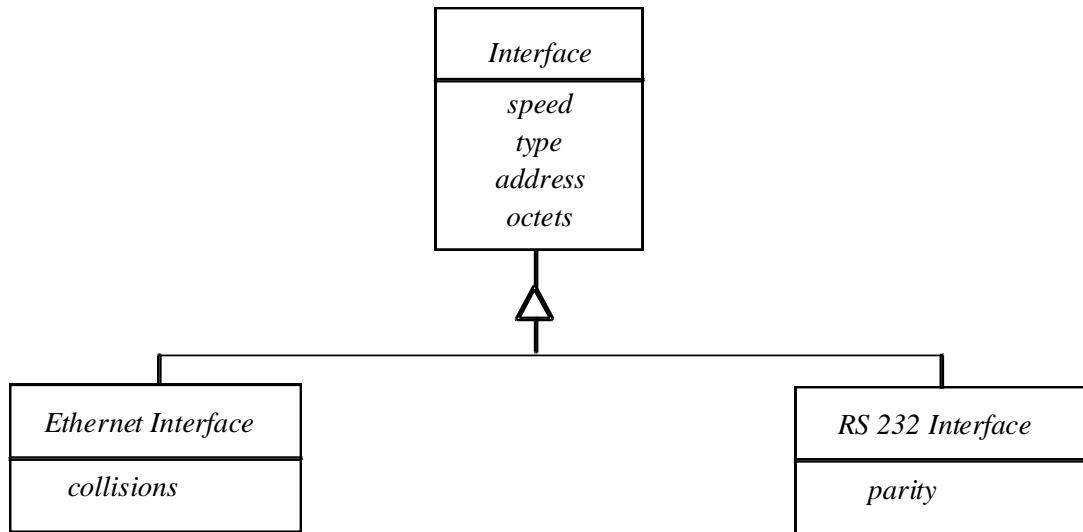


Figure 9.17 Inheritance Used to Define Various Network Interfaces

Notes

- Two options
 - Total object-oriented network management
e.g., XML network management
 - Conversion of SNMP scalar MIB to O-O information modeling
 - e.g. Router with network interface with individual IF MIBs for each technology can be modeled using inheritance classes

Object-Oriented Approach to MIB Engineering (Contd.)

Naming:

Use of MIB group prefix for same name in different subtrees

e.g.: *mib2.system.sysdescr*

mib-2.interfaces.ifTable.ifEntry.ifDescr

Instantiation

```
Router r1, r2; //Statically create 2 routers
```

```
Router *r3;
```

```
..... // Dynamic creation of a router
```

```
if (condition) r3 = new Router;
```

Notes

Object-Oriented Approach to MIB Engineering (Contd.)

- Recommended practices
 - Use of unique prefix: e.g., *if*, *tcp*, etc
 - Extensive use of tables: e.g., *ifTable*, *tcpConnTable*
 - Dynamic creation of rows and *ifNumber*
 - Copy-and-paste of certain blocks of MIB definition from one subtree to another

Notes

MIB ENGINEERING: SMI Tables

- Problem:
 - Indexing of tables to uniquely define a row is complex, especially if it is “multiple-index”
 - Retrieving a row in from a large table is extremely time-consuming and may even time-out
- Solutions to consider in MIB engineering:
 1. Sequence number as index; e.g. port number
 2. Create unique name as index by adding prefix
appTable.appEntry.appPath.'A'. 'p'. 'a'. 'c'. 'h'. 'e'
appTable.appEntry.appPath.65.112.97.99.104.101
 3. OID as Index based on vendor product
 4. Multidimensional tables with multiple index;
Need to select, prioritize and organize the table

Notes

MIB Engineering: SMI Actions

ifAdminStatus OBJECT-TYPE

SYNTAX INTEGER {

up(1), -- ready to pass packets

down(2),

testing(3) -- in some test mode

}

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The desired state of the interface. The testing(3) state indicates that no operational packets can be passed."

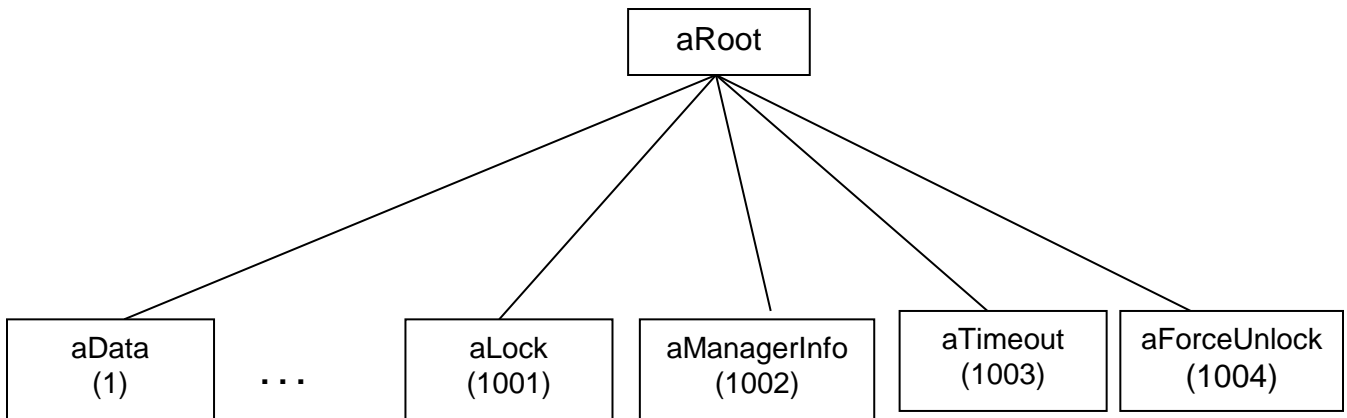
no operational

packets can be passed ::= { ifEntry 7 }

Notes

- Problem: No actions in SNMP defined (as in OSI CMIP/CMISE)
- Solution: Define an action MIB that can be set to perform actions similar to *ifAdminStatus* up(1), down(2), and testing(3)

MIB Engineering: SMI Transactions



Notes

- MIB for process oriented transactions
 - e.g. aRoot above for locking and unlocking data for troubleshooting