# SNMP

Simple Network Management Protocol

# Network Management

- The network management is to
  - Monitor the network
  - Ensure the operations over the network are functional
  - Assure the network works efficiently

- An ounce of prevention is worth a pound of cure
  - Something wrong
    - Service down, fix the problem, resume the service
  - Nothing wrong
    - Service is somewhat abnormal, try to fix it online

- Requirements
  - FCAPS

# Requirements of Network Management

- Fault Management
  - Detect, isolate, reconfigurate and repair the abnormal network environment
  - Problem tracking and control
- Configuration and Name Management
  - Startup, shutdown, reconfigure network component when
    - Upgrade, fault recovery or security checks
- Accounting Management
  - Track the use of network resources by end-user to provide
    - Impropriate usage tracing, charging, statistics
- Performance Management
  - Capacity utilization, throughput, response time, bottleneck
    - Collect information and assess current situation
- Security Management
  - Information protection and access control

# In that time

- Network environment is simple
  - ICMP is the only way to do network investigation
    - ping, traceroute, ....

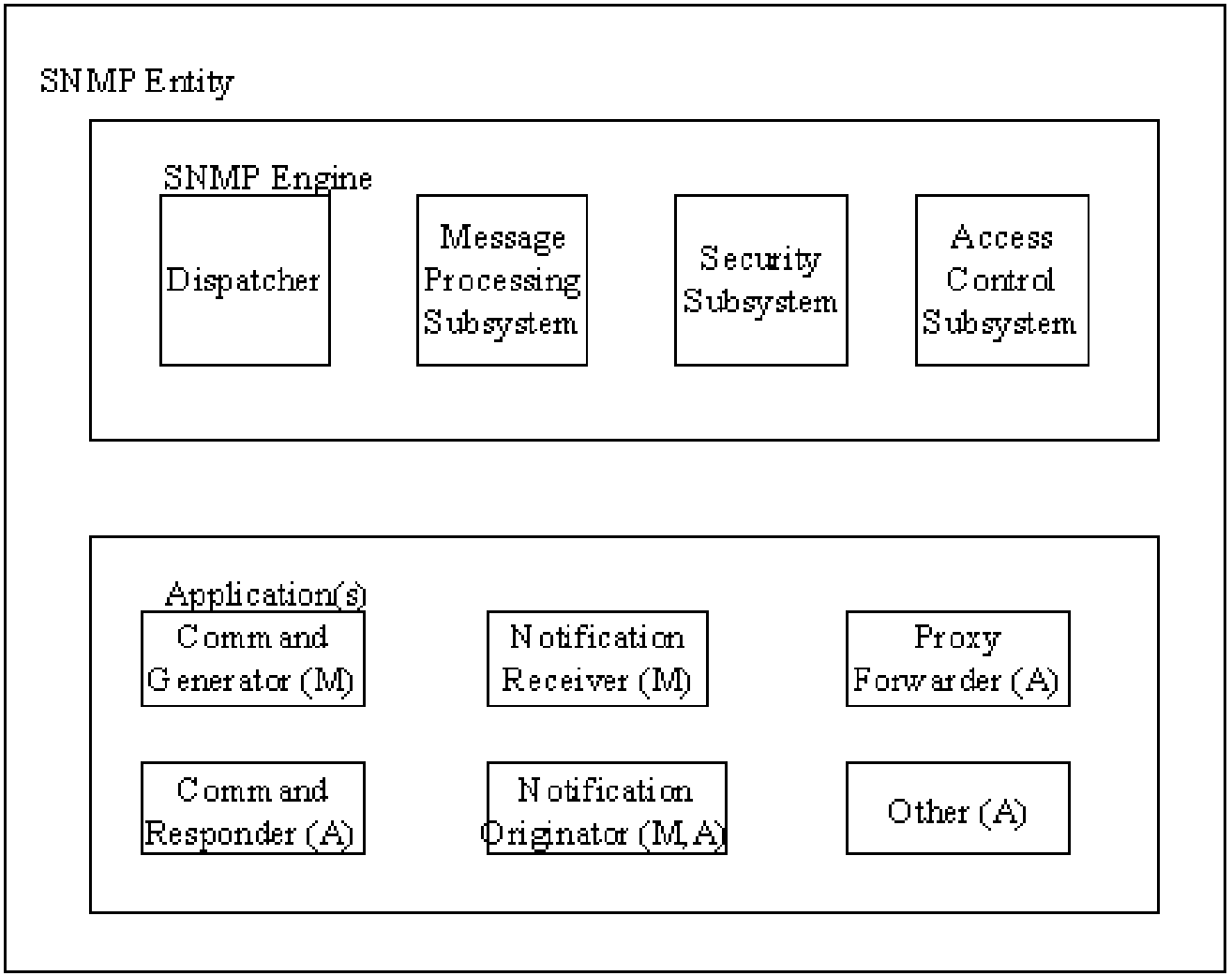
# Introduction

- SNMP – Simple Network Management Protocol
  - A set of standards for network management
    - Protocol
    - Database structure specification
    - Data objects
  - A set of standardized tools that
    - Control costs of network management
    - Across various product types
      - End system, bridges, routers, telecommunications, …
  - Two roles
    - Network management station: SNMP collector, manager
    - SNMP agent

# History

- In 1989
  - SNMP was adopted as TCP/IP-based Internet standards
- In 1991
  - RMON – Remote network MONitoring
    - Supplement to SNMP to include management of LAN and WAN packet flow
- In 1995
  - SNMPv2 (2c)
    - Functional enhancements to SNMP
    - SNMP on OSI-based networks
  - RMON2
    - Network layer and application layer
- In 1998
  - SNMPv3
    - Precise definition, but the content is the same as SNMPv2
    - Security capability for SNMP

# The roles in SNMPv3



# Network Management System (1)

- A collection of tools for
  - Network monitoring
  - Network control
- These tools must be integrated
  - Single operator interface with powerful but user-friendly
  - Support of managed equipments.

# Network Management System (2)

- Architecture of NMS
  - NMA
    - Operator interface
  - NME
    - Collect statistics
    - Response to NMA
    - Alert NMA when environment changing

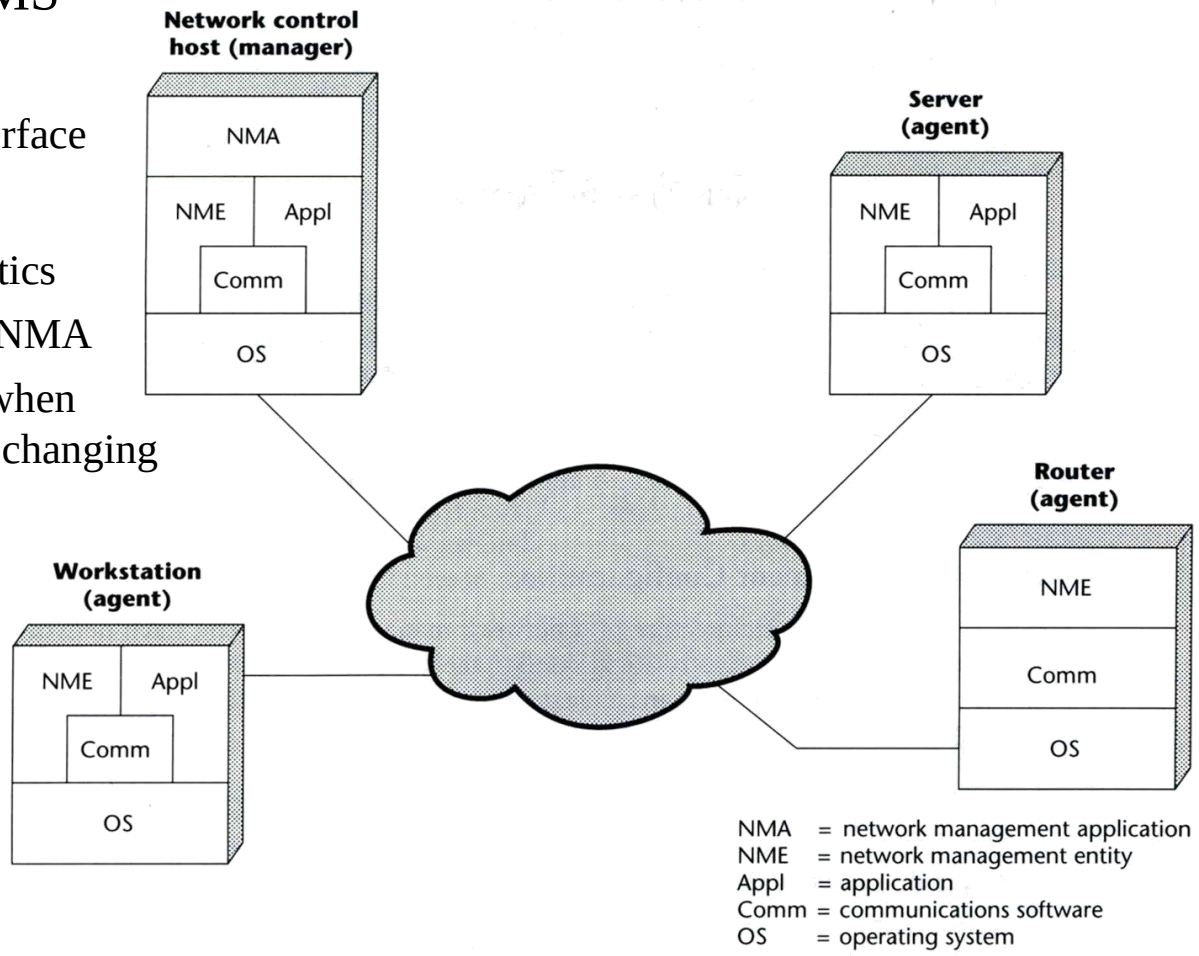



**FIGURE 1.1** Elements of a network management system

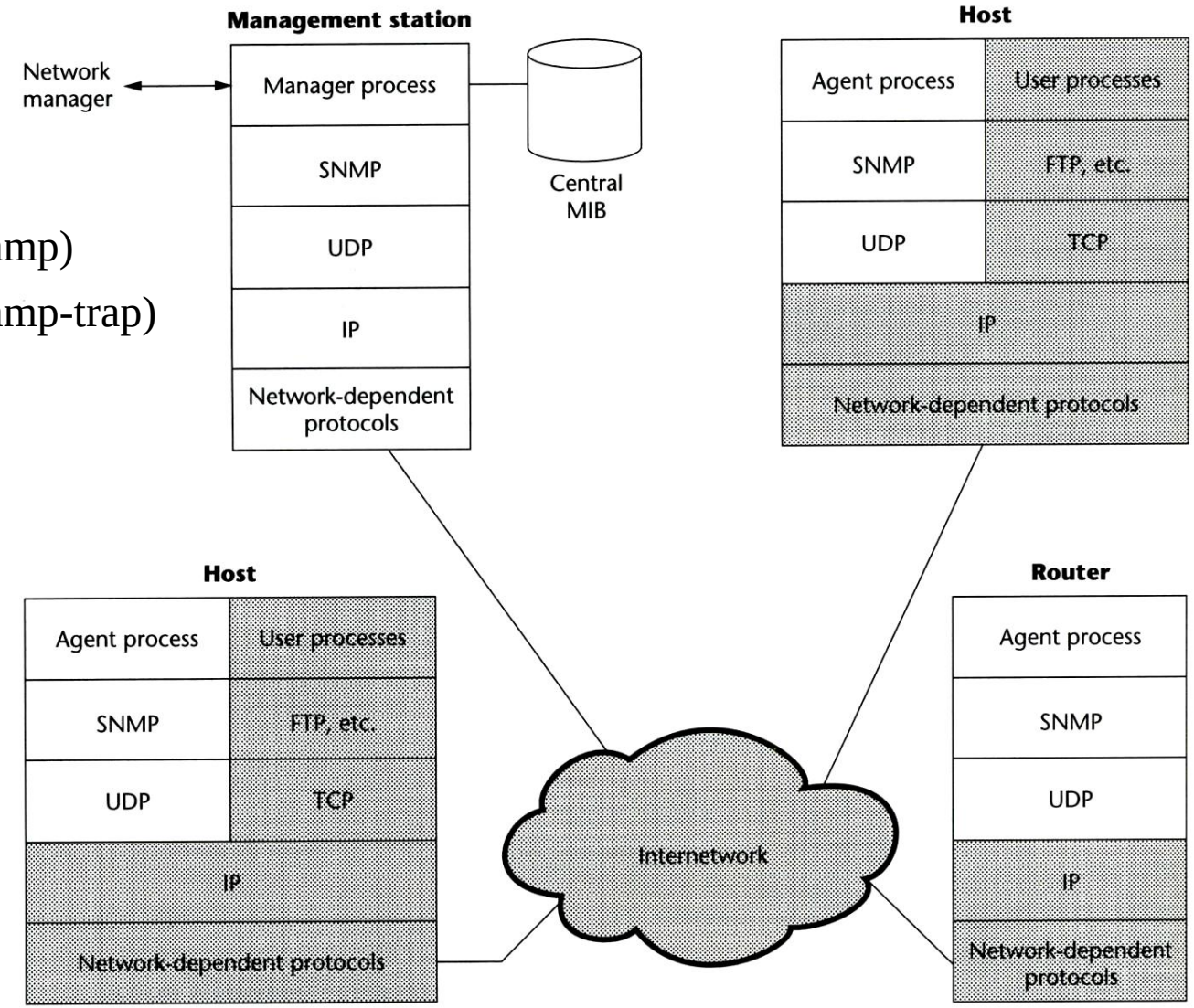
# SNMP Concepts

# SNMP Architecture (1)

- 4 key elements
  - Management station
    - Serve as the interface between manager and devices
      - Management applications
      - User-friendly interface
      - Translate manager's requirements into actual monitoring or control operations
      - Database extracted from MIBs of all managed device
  - Management Agent
    - Respond to request from management station
    - Change settings in MIB of managed device
    - Asynchronously report abnormal event (Trap)
  - Management Information Base (MIB)
    - Each resource is represented as an object and MIB is a collection of objects
  - Network Management Protocol
    - get, setnext, set, getresponse, trap, ...

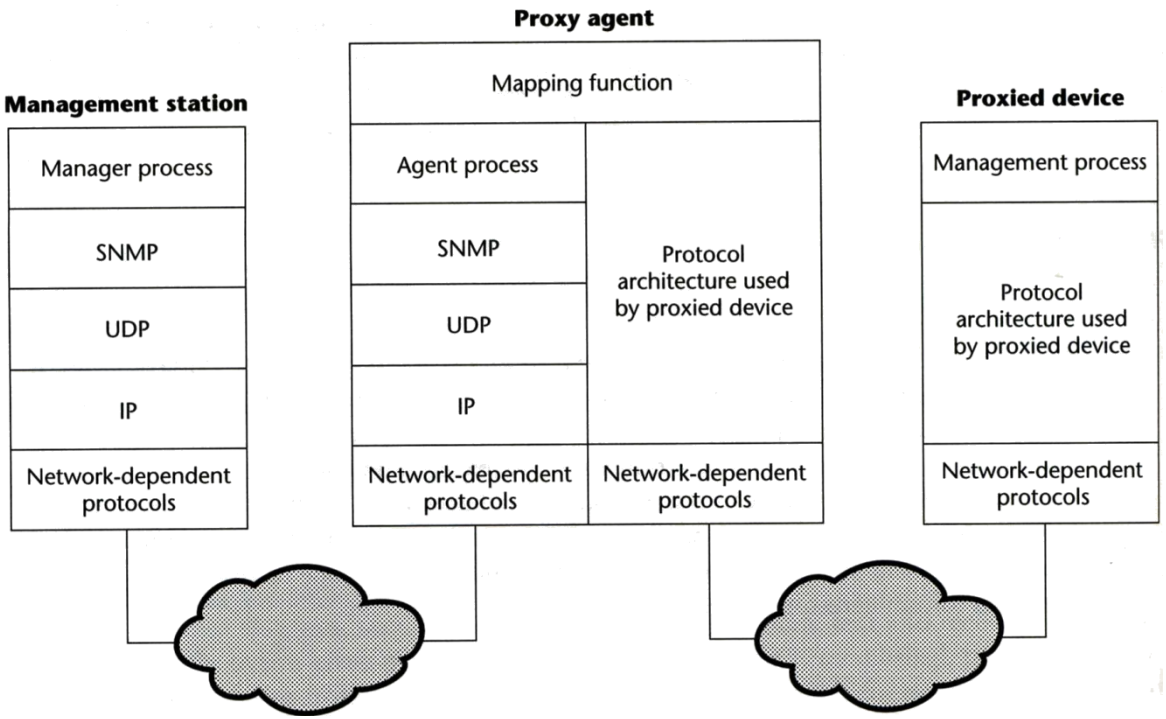
# SNMP Architecture (2)

- SNMP
  - UDP
  - TCP
  - Port 161(snmp)
  - Port 162(snmp-trap)



# SNMP Architecture (3)

- SNMP proxy
  - Devices that do not support UDP/IP
    - ex: Bridge, Modem
  - Devices that do not want to add burden of SNMP agent
    - ex: PC, programmable controller



# SNMP Message Information

- Message Information Base (MIB)
  - Collection of objects
  - Each object represents certain resource of managed device
- Interoperability of MIB
  - Object that represents a particular resource should be the same cross various system
    - What objects
    - (MIB-I) and MIB-II
  - Common representation format
    - SMI (Structure of Management Information)

# SNMP Message Information – SMI (1)

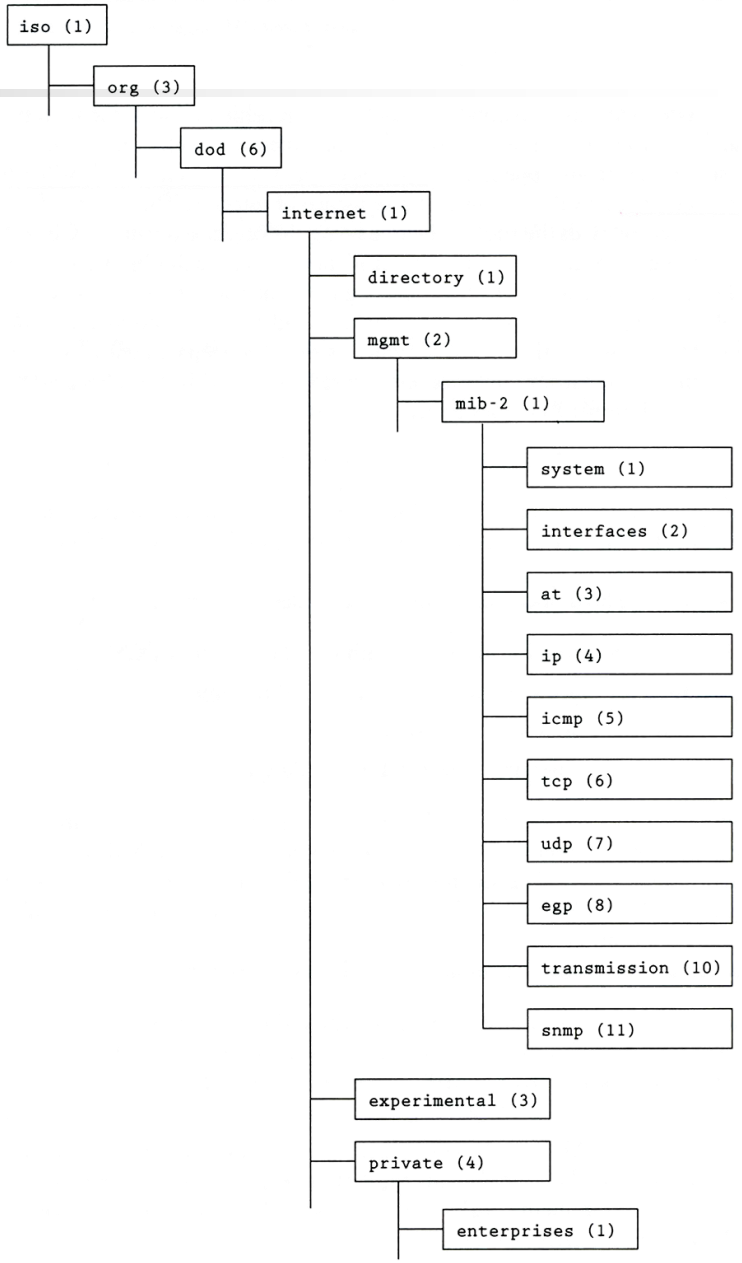
- SMI
  - Structure of Management Information
  - Identify the data type that can be used in MIB
  - How resources are represented and named, including
    - MIB structure
    - Syntax and value of each object
    - Encoding of object value

# SNMP Message Information – SMI (2)

- MIB structure
  - Rooted tree
    - The leaves are the actual managed objects
    - Each object has an identifier (OBJECT IDENTIFIER)
      - Number with dot as delimiter
    - The internet node
      - iso(1) -> org(3) -> dod(6) -> internet(1)
      - object identifier of internet node: 1.3.6.1
    - Under internet node
      - directory(1) :OSI X.500 directory
      - **mgmt(2): used for objects defined in IAB (Internet Activities Board)**
      - experimental(3): used for internet experiments
      - private(4): unilaterally usage

# SNMP Message Information – SMI (3)

- MIB Tree
- Define additional objects
  - Under mib-2
    - 1.3.6.1.2.1
  - Under experimental
    - 1.3.6.1.3
  - Under enterprises
    - 1.3.6.1.4.1



# SNMP Message Information – Object Syntax (1)

- Definition of object
  - Data type
    - Application-independent type (UNIVERSAL type)
      - integer, octetstring, null, object identifier, sequence
    - Application-wide types (RFC 1155)
      - Networkaddress → IP Address
      - counter (0 ~ $2^{32}$ -1), increasing only, wrap to 0
      - gauge (0 ~ $2^{32}$ -1)
      - timeticks
      - opaque (encoded as OCTET STRING for transmission)
      - threshold
  - Value ranges
  - Relationship with other objects in MIB

# SNMP Message Information – Object Syntax (2)

- ASN.1
  - Abstract Syntax Notation One
  - A formal language developed by CCITT and ISO
  - In SNMP, we use macro to define other types used to define managed objects
    - Macro definition (template)
    - Macro instance (particular type)
    - Macro instance value

# SNMP Message Information – Object Syntax (3)

- Example: /usr/share/snmp/mibs/BEGEMOT-HOSTRES-MIB.txt

```
-- Additional stuff for the HOST-RESOURCES MIB.
BEGEMOT-HOSTRES-MIB DEFINITIONS ::= BEGIN

IMPORTS
   MODULE-IDENTITY, OBJECT-TYPE, TimeTicks
      FROM SNMPv2-SMI
   begemot
      FROM BEGEMOT-MIB;

begemotHostres MODULE-IDENTITY
   ....
   ::= { begemot 202 }

begemotHostresObjects   OBJECT IDENTIFIER ::= { begemotHostres 1 }

begemotHrStorageUpdate OBJECT-TYPE
   SYNTAX      TimeTicks
   MAX-ACCESS  read-write
   STATUS      current
   DESCRIPTION
         "The maximum number of ticks the storage table is cached."
::= { begemotHostresObjects 1 }
```

# SNMP Message Information – Object Syntax (4)

- OBJECT-Type macro

```
IMPORTS   ObjectName, Object Syntax FROM RFC-1155-SMI

OBJECT-TYPE MACRO ::=
BEGIN
     TYPE NOTATION ::=   "SYNTAX"   type (TYPE ObjectSyntax)
                         "ACCESS"   Access
                         "STATUS"   Status
                         DescrPart
                         ReferPart
                         IndexPart
                         DefValPart
     VALUE NOTATION ::= value (VALUE ObjectName)

     Access ::= "read-only"|"read-write"|"write-only"|"not-accessible"

     Status ::= "mandatory|"optional"|"obsolete"|"deprecated"

     DescrPart ::= "DESCRIPTION" value (description DisplayString)|empty

     ReferPart ::= "REFERENCE" value (reference DisplayString)|empty

     IndexPart ::= "INDEX" "{" IndexTypes "}"

     IndexTypes ::= IndexType|IndexTypes "," IndexType

     IndexType ::= value (indexobject ObjectName)  --if indexobject, use the SYNTAX
                                                   --value of the correspondent
                                                   --OBJECT-TYPE invocation
                   |type (indextype)               --otherwise use named SMI type;
                                                   --must conform to IndexSyntax below

     DefValPart ::= "DEFVAL" "{" value (defvalue ObjectSyntax) "}" |empty

     DisplayString ::= OCTET STRING SIZE (0..255)

END

IndexSyntax ::= CHOICE { number INTEGER (0..MAX),
                         string OCTET STRING,
                         object OBJECT IDENTIFIER,
                         address NetworkAddress,
                         IpAddress IpAddress }
```

# SNMP Message Information – Object Syntax (5)

- Example of object definition
  - iso.org.dod.internet.mgmt.mib-2.tcp.tcpMaxConn
  - 1.3.6.1.2.1.6.4

```
tcpMaxConn OBJECT-TYPE
     SYNTAX INTEGER
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
         "The limit on the total number of TCP connections the entity can
         support. In entities where the maximum number of connections is
         dynamic, this object should contain the value -1."
     ::= { tcp 4 }
```

# SNMP Message Information – Object Syntax (6)

- 2-D table
  - Two-dimensional array with scalar-valued entries
  - Ex: tcpConnTable (RFC1213)

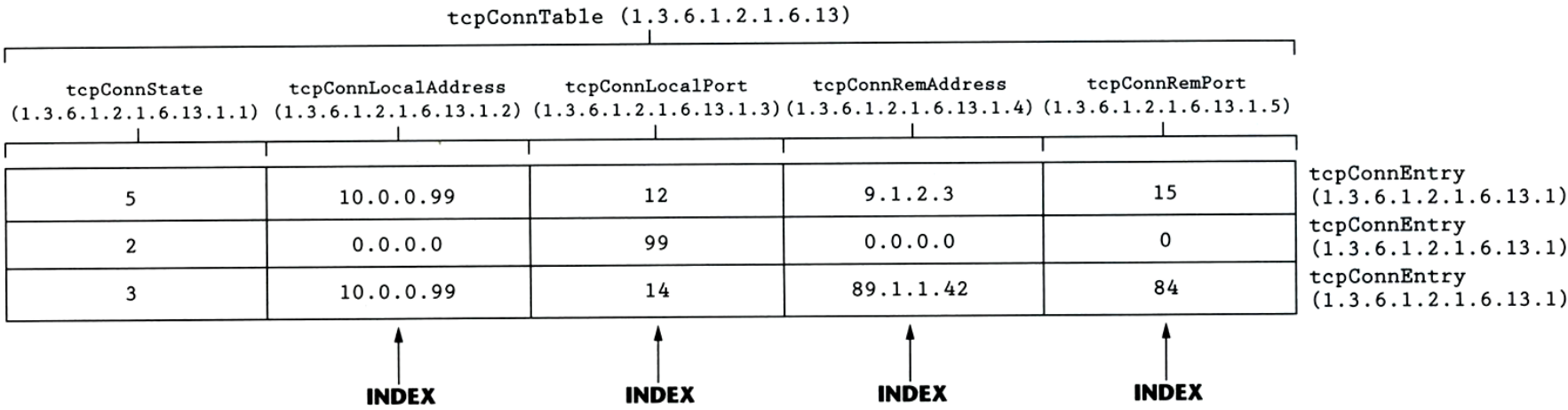
```
tcpConn Table OBJECT-TYPE
        SYNTAX   SEQUENCE OF TcpConnEntry
        ACCESS   not-accessible
        STATUS   mandatory
        DESCRIPTION
              "A table containing TCP connection-specific information."
        ::= { tcp 13 }

tcpConnEntry OBJECT-TYPE
       SYNTAX    TcpConnEntry
       ACCESS    not-accessible
       STATUS    mandatory
       DESCRIPTION
             "Information about a particular TCP connection. An object of this type is
             transient, in that it ceases to exist when (or soon after) the connection
             makes the transition to the CLOSED state."
       INDEX     { tcpConnLocalAddress,
                   tcpConnLocalPort,
                   tcpConnRemAddress,
                   tcpConnRemPort }
       ::= { tcpConnTable 1 }

TcpConnEntry ::= SEQUENCE { tcpConnState INTEGER,
                            tcpConnLocalAddress IpAddress,
                            tcpConnLocalPort INTEGER (0..65535),
                            tcpConnRemAddress IpAddress
                            tcpConnRemPort INTEGER (0..65535)}
```

# SNMP Message Information – Object Syntax (7)

```
tcpConnState OBJECT-TYPE
      SYNTAX   INTEGER {closed (1),
                        listen (2),
                        synSent (3),
                        synReceived (4),
                        established (5),
                        finWait1 (6),
                        finWait2 (7),
                        closeWait (8),
                        lastAck (9),
                        closing (10),
                        timeWait (11),
                        delete TCB (12) }

      ACCESS      read-write
      STATUS      mandatory
      DESCRIPTION
            "The state of this TCP connection.
            ::= { tcpConnEntry 1 }

tcpConnLocalAddress OBJECT-TYPE
      SYNTAX      IpAddress
      ACCESS      read-only
      STATUS      mandatory
      DESCRIPTION
            "The local IP address for this TCP connection. In the case of a connec-
            tion in the listen state which is willing to accept connections for any
            IP interface associated with the node, the value 0.0.0.0 is used."
      ::= { tcpConnEntry 2 }

tcpConnLocalPort OBJECT-TYPE
      SYNTAX      INTEGER (0..65535)
      ACCESS      read-only
      STATUS      mandatory
      DESCRIPTION
            "The local port number for this TCP connection."
      ::= { tcpConnEntry 3 }

tcpConnRemAddress OBJECT-TYPE
      SYNTAX      IpAddress
      ACCESS      read-only
      STATUS      mandatory
      DESCRIPTION
            "The remote IP address for this TCP connection."
      ::= { tcpConnEntry 4 }

tcpConnRemPort OBJECT-TYPE
      SYNTAX      INTEGER (0..65535)
      ACCESS      read-only
      STATUS      mandatory
      DESCRIPTION
            "The remote port number for this TCP connection."
      ::= { tcpConnEntry 5 }
```

# SNMP Message Information – Object Syntax (8)

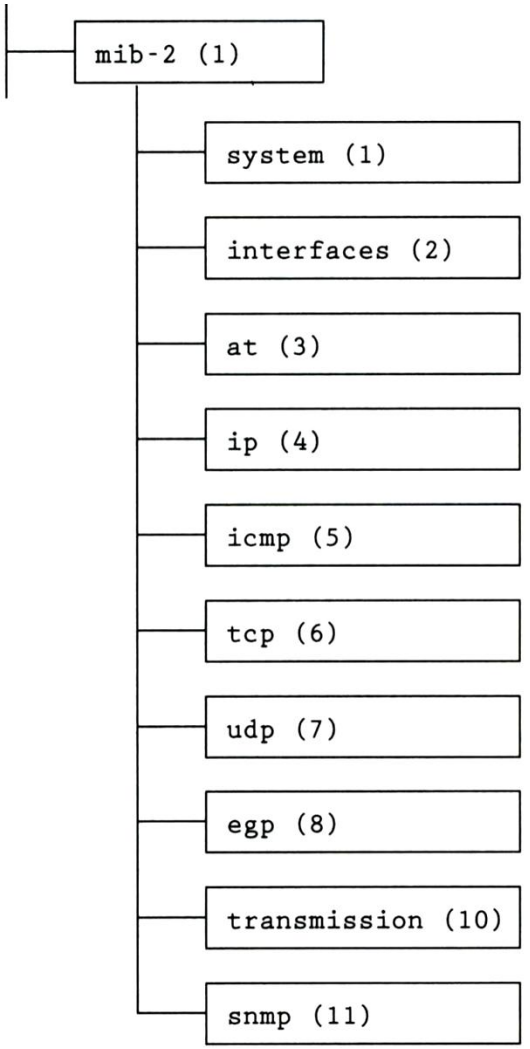
- iso (1) -> org (3) -> dod (6) -> internet (1) -> mgmt (2)
  - mib-2 (1) -> tcp (6) -> tcpConnTable(13)



# Standard MIBs

# MIB-II (1)

- RFC1213
  - MIB-I (RFC 1156)
  - MIB-II is a superset of MIB-I with some additional objects and groups

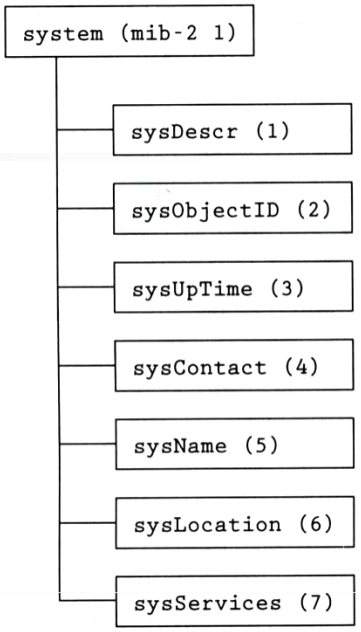


# MIB-II (2)

- First layer under mib-2
  - 1.3.6.1.2.1 (iso.org.dod.internet.mgmt.mib-2)
  - system
    - Overall information about the system
  - interfaces
    - Information about each interface
  - at
    - Address translation (obsolete)
  - ip, icmp, tcp, udp, egp
  - transmission
    - Transmission schemes and access protocol at each system interface
  - snmp

# MIB-II

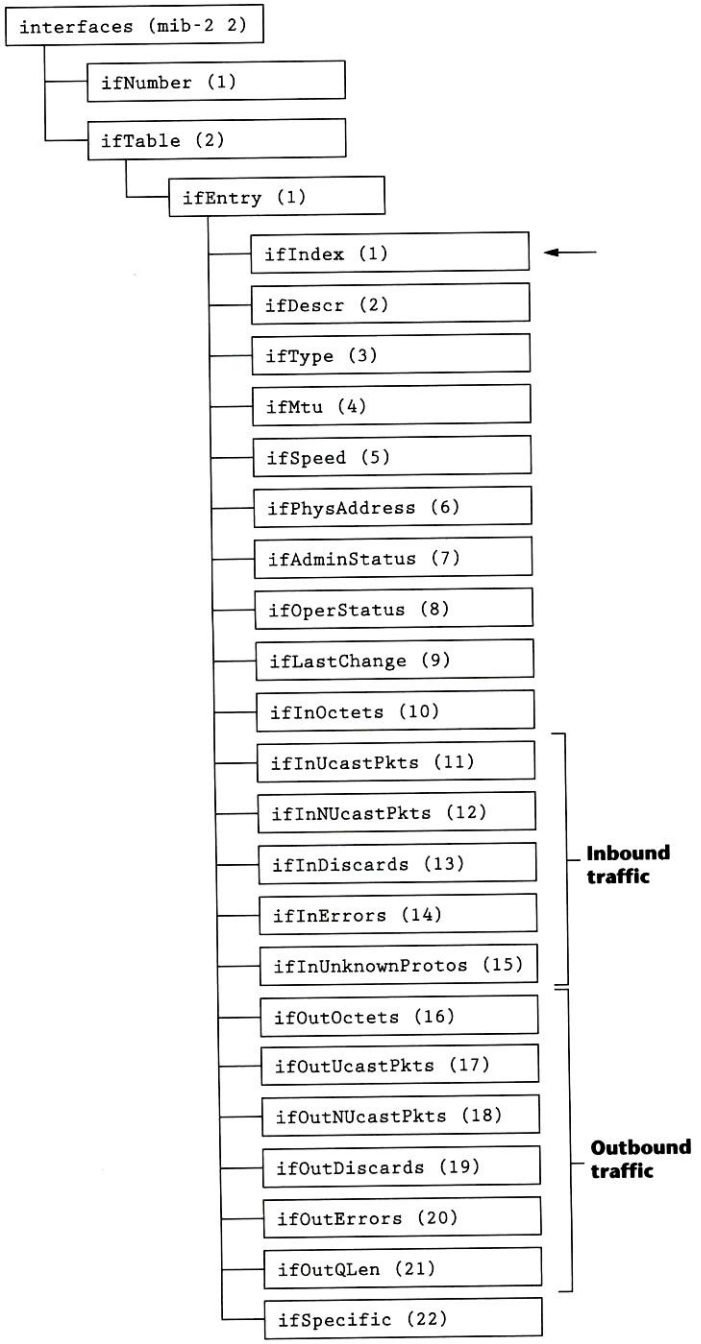
## system group




- sysServices
  - 1 physical (ex: repeater)
  - 2 datalink/subnetwork (ex: bridge)
  - 3 internet (ex: router)
  - 4 end-to-end (ex: IP hosts)
  - 7 applications (ex: mail relays)

| Object | Syntax | Access | Description |
|---|---|---|---|
| sysDescr | DisplayString (SIZE (0 . . . 255)) | RO | A description of the entity, such as hardware, operating system, etc. |
| sysObjectID | OBJECT IDENTIFIER | RO | The vendor's authoritative identification of the network management subsystem contained in the entity |
| sysUpTime | TimeTicks | RO | The time since the network management portion of the system was last reinitialized |
| sysContact | DisplayString (SIZE (0 . . . 255)) | RW | The identification and contact information of the contact person for this managed node |
| sysName | DisplayString (SIZE (0 . . . 255)) | RW | An administratively assigned name for this managed node |
| sysLocation | DisplayString (SIZE (0 . . . 255)) | RW | The physical location of this node |
| sysServices | INTEGER (0 . . . 127) | RO | A value that indicates the set of services this entity primarily offers |

# MIB-II
## interface group (1)



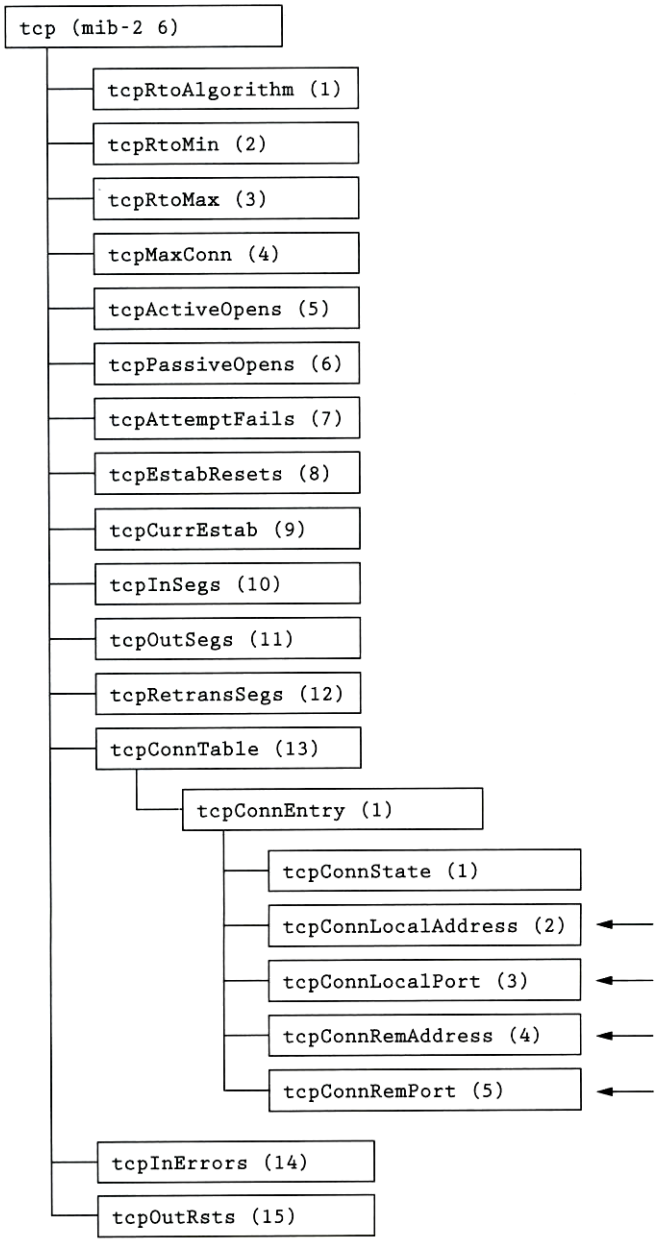
# MIB-II
## interface group (2)

**TABLE 6.2** interfaces Group Objects

| Object | Syntax | Access | Description |
|---|---|---|---|
| ifNumber | INTEGER | RO | The number of network interfaces |
| ifTable | SEQUENCE OF ifEntry | NA | A list of interface entries |
| ifEntry | SEQUENCE | NA | An interface entry containing objects at the subnetwork layer and below for a particular interface |
| ifIndex | INTEGER | RO | A unique value for each interface |
| ifDescr | DisplayString (SIZE (0 . . . 255)) | RO | Information about the interface, including name of manufacturer, product name, and version of the hardware interface |
| ifType | INTEGER | RO | Type of interface, distinguished according to the physical/link protocol(s) |
| ifMtu | INTEGER | RO | The size of the largest protocol data unit, in octets, that can be sent/received on the interface |
| ifSpeed | Gauge | RO | An estimate of the interface's current data rate capacity |
| ifPhysAddress | PhysAddress | RO | The interface's address at the protocol layer immediately below the network layer |
| ifAdminStatus | INTEGER | RW | Desired interface state (up(1), down(2), testing(3)) |
| ifOperStatus | INTEGER | RO | Current operational interface state (up(1), down(2), testing(3)) |
| ifLastChange | TimeTicks | RO | Value of sysUpTime at the time the interface entered its current operational state |
| ifInOctets | Counter | RO | Total number of octets received on the interface, including framing characters |
| ifInUcastPkts | Counter | RO | Number of subnetwork-unicast packets delivered to a higher-layer protocol |
| ifInNUcastPkts | Counter | RO | Number of nonunicast packets delivered to a higher-layer protocol |
| ifInDiscards | Counter | RO | Number of inbound packets discarded, even though no errors had been detected, to prevent their being deliverable to a higher-layer protocol (e.g., buffer overflow) |
| ifInErrors | Counter | RO | Number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol |
| ifInUnknownProtos | Counter | RO | Number of inbound packets that were discarded because of an unknown or unsupported protocol |
| ifOutOctets | Counter | RO | Total number of octets transmitted on the interface, including framing characters |
| ifOutUcastPkts | Counter | RO | Total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or otherwise not sent |
| ifOutNUcastPkts | Counter | RO | Total number of packets that higher-level protocols requested be transmitted to a nonunicast address, including those that were discarded or otherwise not sent |
| ifOutDiscards | Counter | RO | Number of outbound packets discarded even though no errors had been detected to prevent their being transmitted (e.g., buffer overflow) |
| ifOutErrors | Counter | RO | Number of outbound packets that could not be transmitted because of errors |
| ifOutQLen | Gauge | RO | Length of the output packet queue |
| ifSpecific | OBJECT IDENTIFIER | RO | Reference to MIB definitions specific to the particular media being used to realize the interface |

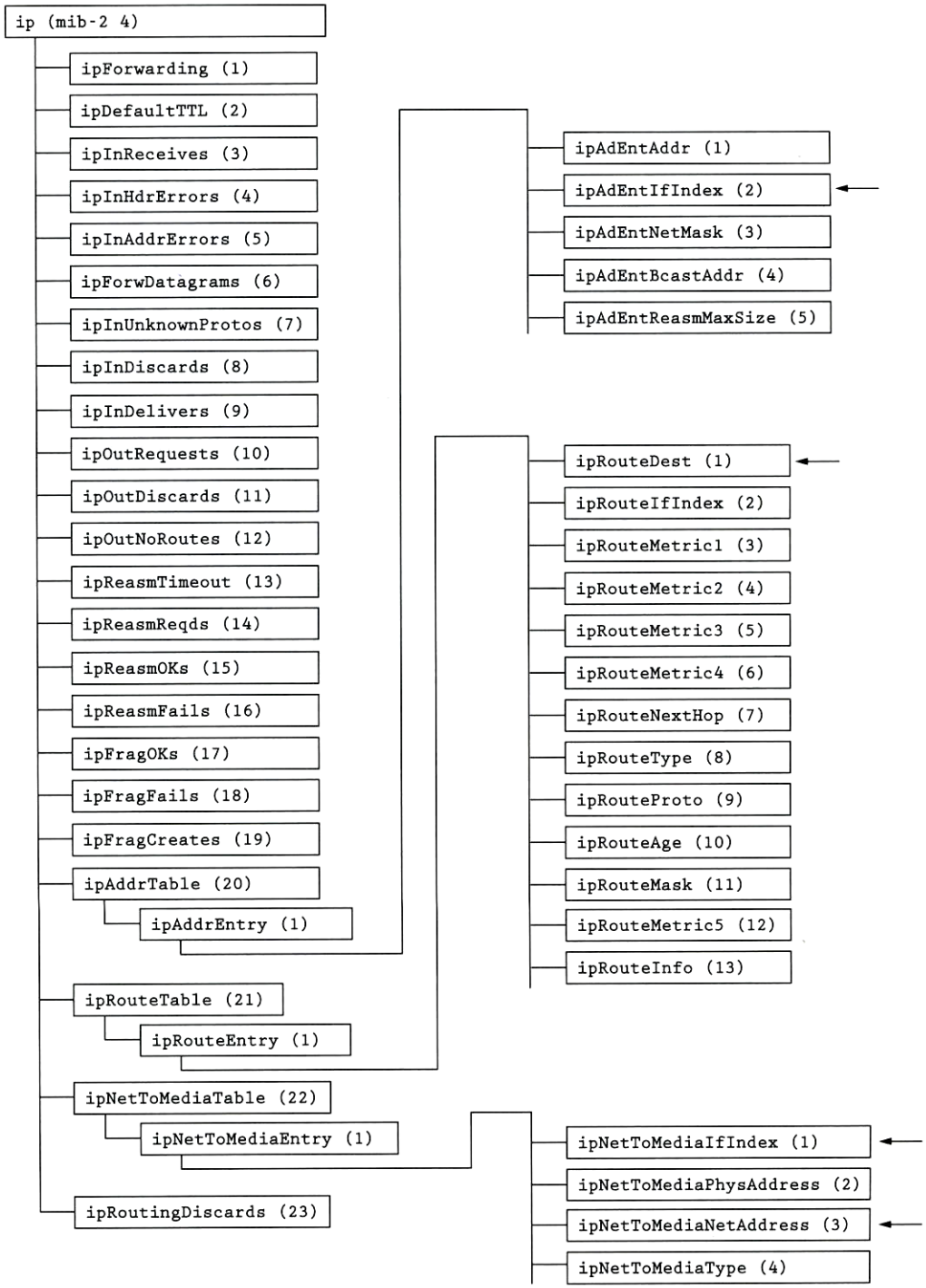
# MIB-II

## tcp group



# MIB-II

## ip group



# Host Resource MIB

- RFC2790
  - host OBJECT IDENTIFIER ::= { mib-2 25 }
  - hrSystem OBJECT IDENTIFIER ::= { host 1 }
  - hrStorage OBJECT IDENTIFIER ::= { host 2 }
  - hrDevice OBJECT IDENTIFIER ::= { host 3 }
  - hrSWRun OBJECT IDENTIFIER ::= { host 4 }
  - hrSWRunPerf OBJECT IDENTIFIER ::= { host 5 }
  - hrSWInstalled OBJECT IDENTIFIER ::= { host 6 }
  - hrMIBAdminInfo OBJECT IDENTIFIER ::= { host 7 }

# SNMP Protocol

# SNMP Protocol

- Supported operations
  - get, getnext, set, getresponse, trap, …
- Simplicity vs. limitations
  - Not possible to change the structure of MIB by adding or deleting object instances
  - Access is provided only to leaf objects

# SNMP Protocol – security concern

- In management environment
  - The management station and managed agent
    - One-to-many relationship
    - One station may manage all or a subset of target
  - The managed station and management station
    - One-to-many relationship
    - Each managed agent controls its local MIB and must be able to control the use of that MIB
    - Three aspects
      - Authentication service
      - Access policy
      - Proxy service

# SNMP Protocol – communities (1)

- An SNMP community
  - A relationship between an SNMP agent and a set of SNMP managers that defines
    - Authentication, access control and proxy
  - The managed system establishes one community for each combination of authentication, access control and proxy
  - Each community has a unique “community name”
  - Management station use certain community name in all get and set operations

# SNMP Protocol – communities (2)

- Authentication
  - The community name (password)
- Access policy
  - Community profile
    - SNMP MIB view
      - A subset of MIB objects
    - SNMP access mode
      - read-only, read-write, write-only, non-accessible

# SNMP Protocol – Where is the security

- SNMPv3
  - User-based Security Model (USM)
    - Message Authentication
      - HMAC
        - MD5, SHA-1
        - Authentication passphrase, secret key
    - Encryption
      - CBC-DES
  - View-based Access Control Model (VACM)
    - Context table
    - Security to group table
    - Access table
    - View tree family table

# Net-SNMP

previously known as "ucd-snmp"

# Net-SNMP (1)

- Install net-snmp
  - net-mgmt/net-snmp
  - # make [OPTIONS] install clean

```
DEFAULT_SNMP_VERSION="3"      Default version of SNMP to use.
NET_SNMP_SYS_CONTACT="nobody@nowhere.invalid"
                              Default system contact.
NET_SNMP_SYS_LOCATION="somewhere"
                              Default system location.
NET_SNMP_LOGFILE="/var/log/snmpd.log"
                              Default log file location for snmpd.
NET_SNMP_PERSISTENTDIR="/var/net-snmp"
                              Default directory for persistent data storage.
```

  - Firewall allows
    - snmpd: udp 161
    - snmptrapd: udp 162

# Net-SNMP (2)

- After installing…

```
If you want to invoke snmpd and/or snmptrapd at startup, put these
    lines into /etc/rc.conf.

      snmpd_enable="YES"
      snmpd_flags="-a"
      snmpd_conffile="/usr/local/share/snmpd.conf /etc/snmpd.conf"
      snmptrapd_enable="YES"
      snmptrapd_flags="-a -p /var/run/snmptrapd.pid"
```

- /usr/local/share/snmp/snmpd.conf.example

```
                                    #  Full access from the local host
# rocommunity public  localhost
                                    #  Default access to basic system info
rocommunity public  default    -V systemonly
```

# Net-SNMP (3)

- Use snmpconf command to generate the configuration files
  - snmpconf -g basic_setup
  - snmpconf
    - System Information Setup
      - Location, contact, service
    - Access Control Setup
      - SNMPv3 or SNMPv1 access community
    - Trap Destination
      - Where to send the trap
    - Monitor Various Aspects of the Running Host
      - Process, disk space, load, file
    - Extending the Agent
      - Let snmp agent to return information that yourself define
    - Agent Operating Mode
      - User/group, IP port,...

# Net-SNMP (4)

- To get various value
  - man snmpget, snmpgetnext, snmptable

```
% snmpget -c public -v 1 nasa system.sysContact.0
% snmpgetnext −c public −v 1 nasa system.sysContact.0
% snmptable -c public -v 1 nasa mib-2.tcp.tcpConnTable
% snmpwalk −c public −v 1 nasa system
% snmpwalk -c public -v 1 nasa iso.org.dod.internet.private.enterprises
```

# RRDTool

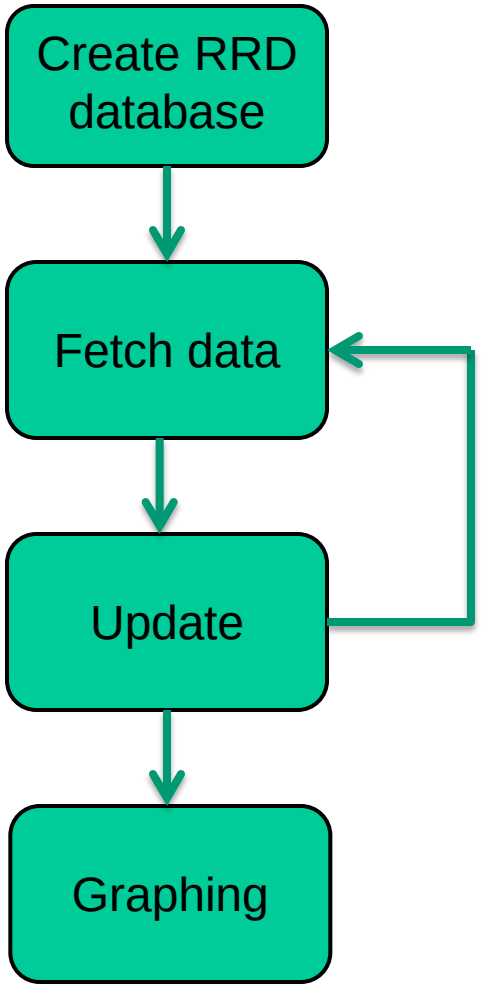
# RRDTool (1)

- Round-robin database.
- Data logging / graphing.
- Easy integration.
  - Bindings for Python, Perl, Ruby, Lua, ...
  - Command line tools also enable integration with shell scripts.

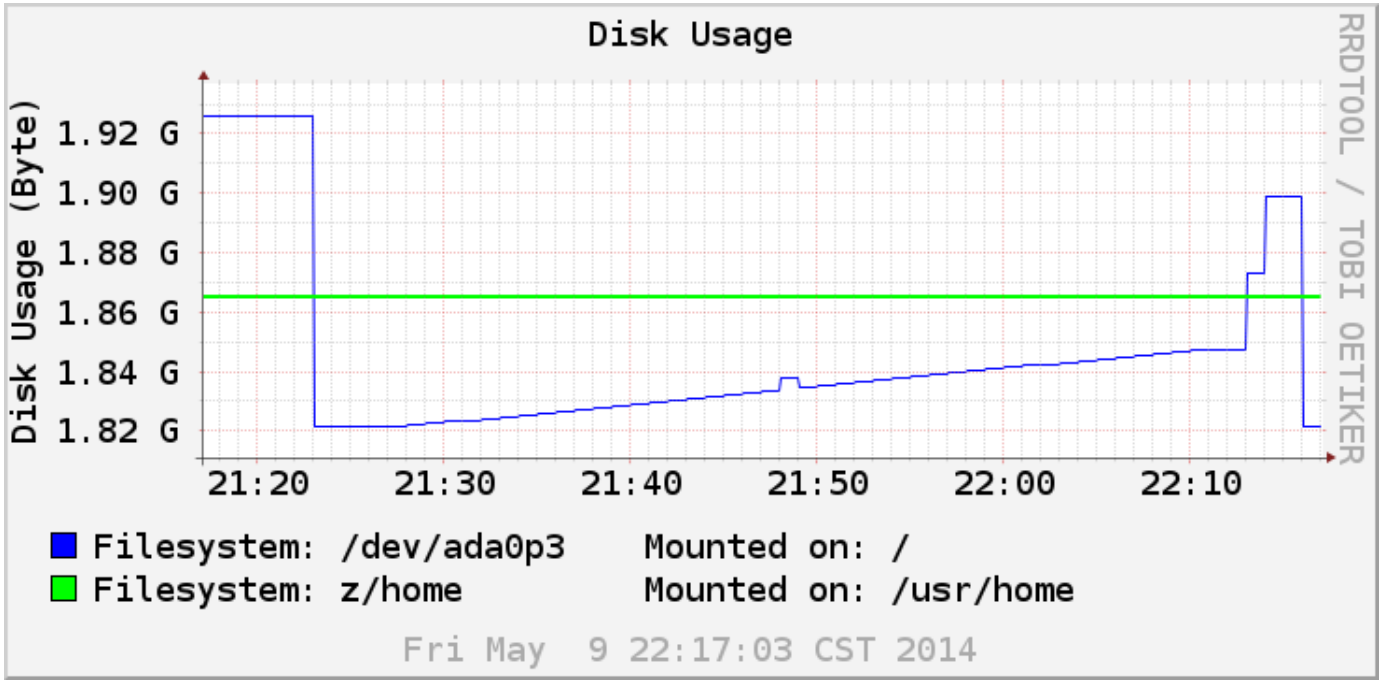
- FreeBSD port: databases/rrdtool
- http://oss.oetiker.ch/rrdtool/doc/index.en.html

# RRDTool (2)

- Create
  - Create new RRD database.
  - $ rrdtool create
- Fetch data
- Update
  - Update the database.
  - $ rrdtool update
- Graph
  - Graph the result from database.
  - $ rrdtool graph



# RRDTool (3)


Disk Usage
Disk Usage (Byte)
1.92 G
1.90 G
1.88 G
1.86 G
1.84 G
1.82 G
21:20
21:30
21:40
21:50
22:00
22:10
Filesystem: /dev/ada0p3 Mounted on: /
Filesystem: z/home Mounted on: /usr/home
Fri May 9 22:17:03 CST 2014
RRDTOOL / TOBI OETIKER

# Cacti

# Cacti(1)

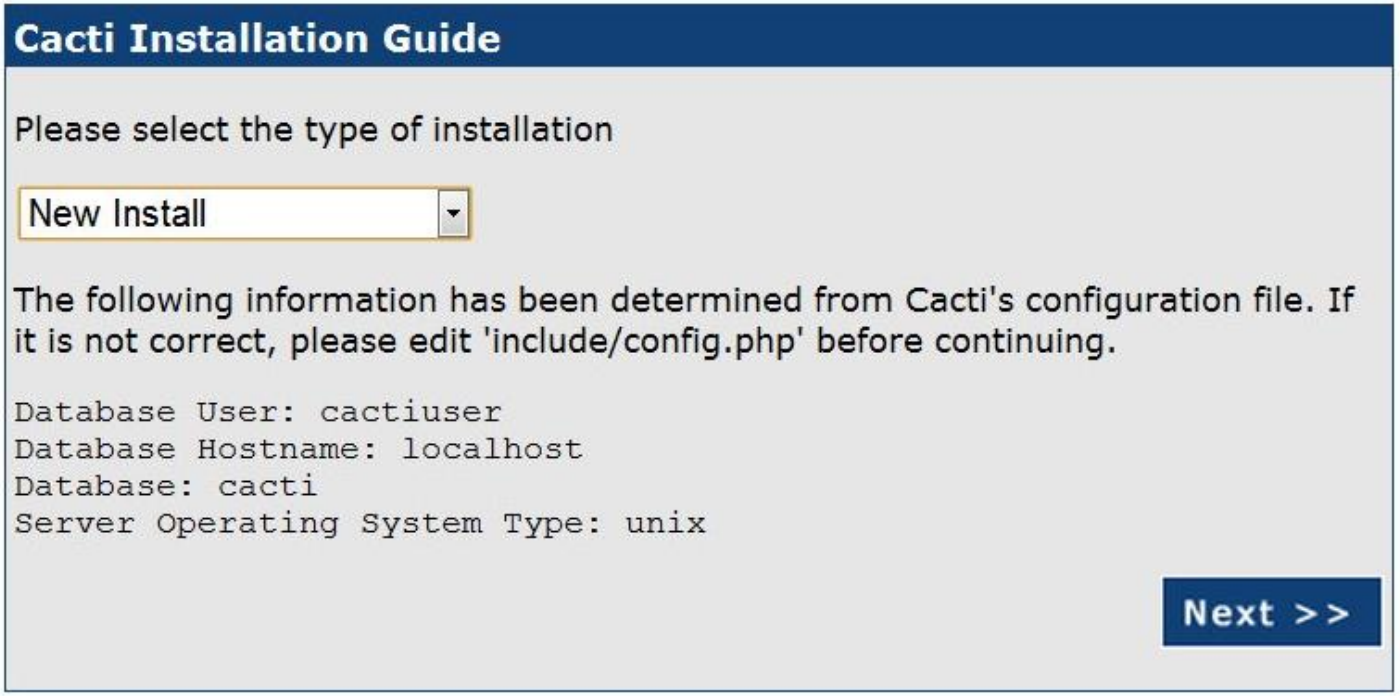
- About
  - Cacti is a complete network graphing solution designed to harness the power of RRDTool's data storage and graphing functionality.
  - Cacti provides a fast poller, advanced graph templating, multiple data acquisition methods, and user management features out of the box.
  - All of this is wrapped in an intuitive, easy to use interface that makes sense for LAN-sized installations up to complex networks with hundreds of devices.

- Install cacti
  - /usr/ports/net-mgmt/cacti

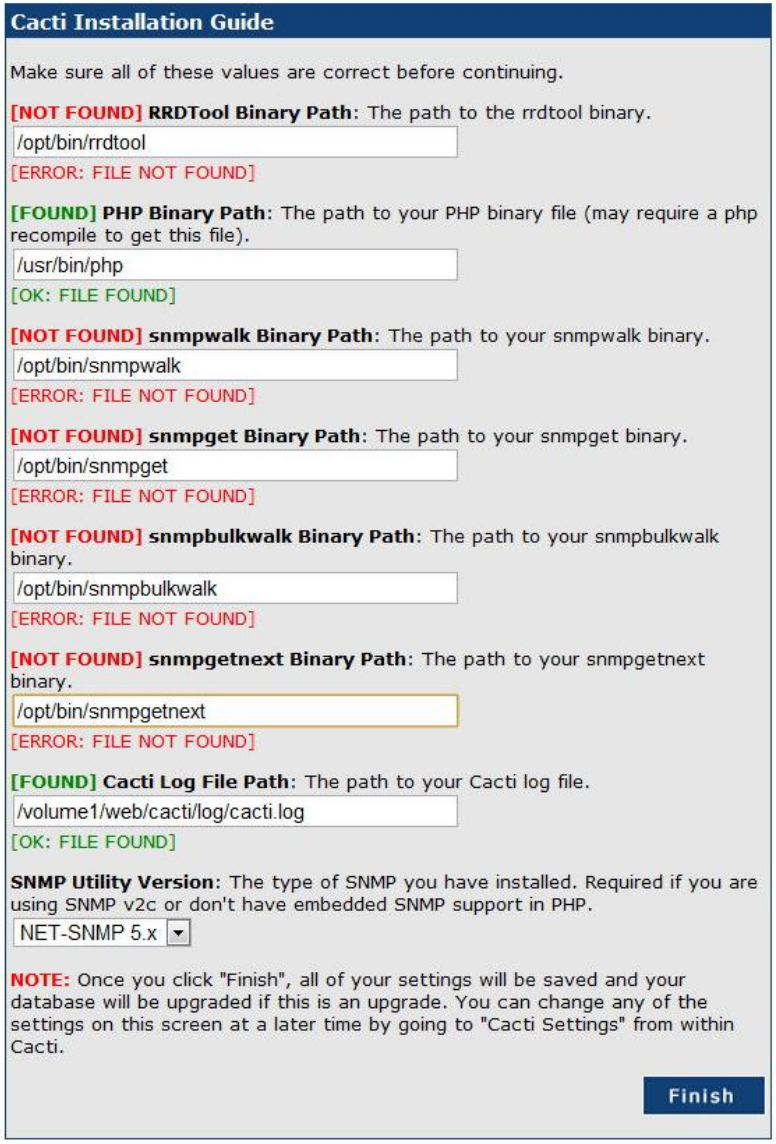
# Cacti(2)

```
└─[~] >pkg info -D cacti
cacti-0.8.8b_2:
==========================================================================
Cacti is now installed. If you intall it for the first time,
you may have to follow this steps to make it work correctly:

1. Create the MySQL database:
# mysqladmin --user=root create cacti
2. Create a mysql user/password for cacti:
   (change user and/or password if required)
# echo "GRANT ALL ON cacti.* TO cactiuser@localhost IDENTIFIED BY 'cactiuser'; FLUSH
 PRIVILEGES;" | mysql
3. Import the default cacti database:
# mysql cacti < /usr/local/share/cacti/cacti.sql
4. Edit /usr/local/share/cacti/include/config.php.
5. Add the line to cron jobs with the command:
# crontab -u cacti -e
*/5 * * * * /usr/local/bin/php /usr/local/share/cacti/poller.php > /dev/null 2>&1
6. Add alias in apache config for the cacti dir:
Alias /cacti "/usr/local/share/cacti/"
7. Be sure apache gives an access to the directory ('Allow from' keywords).
8. Open a cacti login page in your web browser and login with admin/admin.

If you update cacti, open a login page, an updating process
will start automatically.
==========================================================================
```

# Cacti(3)


Cacti Installation Guide
Please select the type of installation
New Install
The following information has been determined from Cacti's configuration file. If it is not correct, please edit 'include/config.php' before continuing.
Database User: cactiuser
Database Hostname: localhost
Database: cacti
Server Operating System Type: unix
Next >>

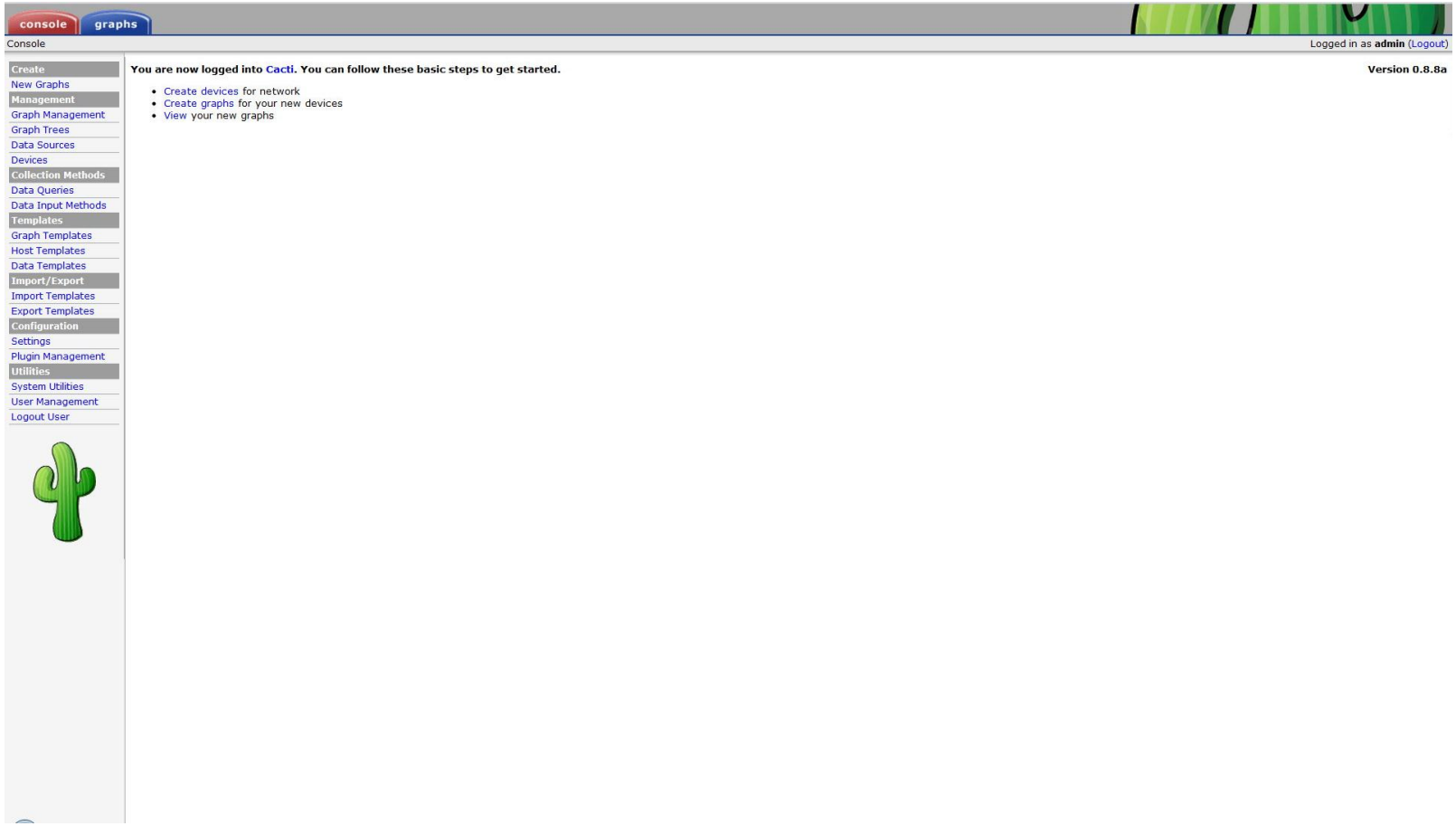
# Cacti(4)


Cacti Installation Guide
Make sure all of these values are correct before continuing.
[NOT FOUND] RRDTool Binary Path: The path to the rrdtool binary.
/opt/bin/rrdtool
[ERROR: FILE NOT FOUND]
[FOUND] PHP Binary Path: The path to your PHP binary file (may require a php recompile to get this file).
/usr/bin/php
[OK: FILE FOUND]
[NOT FOUND] snmpwalk Binary Path: The path to your snmpwalk binary.
/opt/bin/snmpwalk
[ERROR: FILE NOT FOUND]
[NOT FOUND] snmpget Binary Path: The path to your snmpget binary.
/opt/bin/snmpget
[ERROR: FILE NOT FOUND]
[NOT FOUND] snmpbulkwalk Binary Path: The path to your snmpbulkwalk binary.
/opt/bin/snmpbulkwalk
[ERROR: FILE NOT FOUND]
[NOT FOUND] snmpgetnext Binary Path: The path to your snmpgetnext binary.
/opt/bin/snmpgetnext
[ERROR: FILE NOT FOUND]
[FOUND] Cacti Log File Path: The path to your Cacti log file.
/volume1/web/cacti/log/cacti.log
[OK: FILE FOUND]
SNMP Utility Version: The type of SNMP you have installed. Required if you are using SNMP v2c or don't have embedded SNMP support in PHP.
NET-SNMP 5.x
NOTE: Once you click "Finish", all of your settings will be saved and your database will be upgraded if this is an upgrade. You can change any of the settings on this screen at a later time by going to "Cacti Settings" from within Cacti.
Finish

# Cacti(5)

- Default account/pwd
  - admin/admin

# Cacti(6)


console
graphs
Console
Logged in as admin (Logout)
Create
New Graphs
Management
Graph Management
Graph Trees
Data Sources
Devices
Collection Methods
Data Queries
Data Input Methods
Templates
Graph Templates
Host Templates
Data Templates
Import/Export
Import Templates
Export Templates
Configuration
Settings
Plugin Management
Utilities
System Utilities
User Management
Logout User
You are now logged into Cacti. You can follow these basic steps to get started.
Version 0.8.8a
Create devices for network
Create graphs for your new devices
View your new graphs

# Cacti(7)

router - Traffic - Ethernet1/4 Total
bytes per second
3.0 M
2.0 M
1.0 M
0.0
Thu 00:00
Thu 12:00
readynas - Used Space - /c
bytes
800 G
600 G
400 G
200 G
0
synology - Used Space - /volume1
1000 G
synology - Traffic - eth0
5.0 M
4.0 M
readynas - Traffic - eth0
readynas - CPU Usage
percent
10
synology - CPU Usage
60
40
20
readynas - Memory Usage
1.0 G
0.5 G
RRDTOOL / TOBI OETIKER