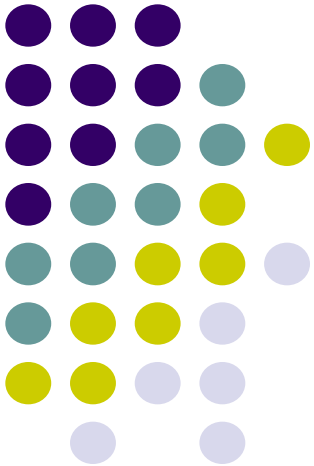
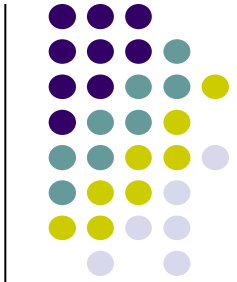
SNMP Tutorial

Dr Pipat Sookavatana

.

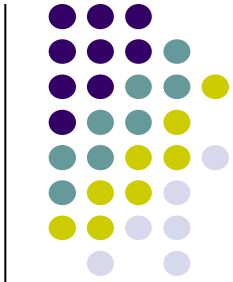


Tutorial Overview



- Introduction
- Management Information Base (MIB)
- Simple Network Management Protocol (SNMP)
- SNMP Commands
- Tools
  - ‘SNMPwalk’ (CLI)
  - ‘MIB Browser’ (GUI)

Introduction



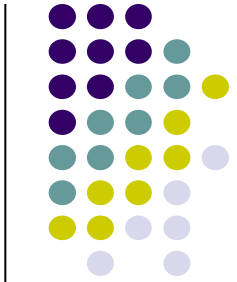
SNMP *Simple Network Management Protocol* is an application layer protocol that facilitates the exchange of management information between network devices

- Application-layer protocol for managing TCP/IP based networks.

- Runs over UDP, which runs over IP using Port 161 and 162

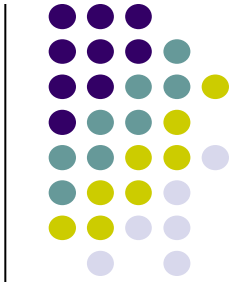
- Two versions of SNMP exist: SNMP version 1 (SNMPv1) and SNMP version 2 (SNMPv2).

Basic tasks that fall under this category are



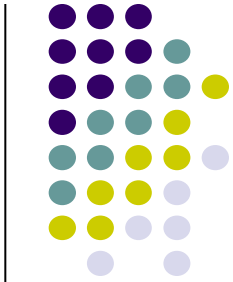
- Configuration Management
  - Keeping track of device setting
- Fault Management
  - -Dealing with problems and emergencies in the network i.e. server, router
- Performance Management

Network Management Success factors



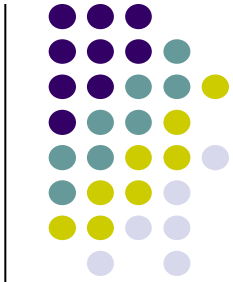
- The management interface must be
  - Standardized
  - Extendable
  - Portable
- The management mechanism must be
  - In expensive

Major functions



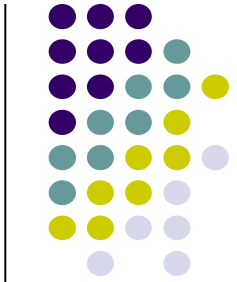
- Configuration Management - inventory, configuration, provisioning
- Fault Management - reactive and proactive network fault management
- Performance Management - # of packets dropped, timeouts, collisions, CRC errors
- Security Management - SNMP doesn't provide much here
- Accounting Management - cost management and chargeback assessment
- Asset Management - statistics of equipment, facility, and administration personnel
- Planning Management - analysis of trends to help justify a network upgrade or bandwidth increase

History



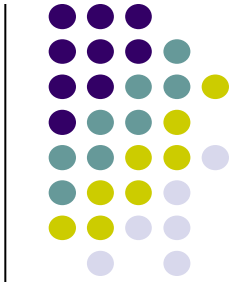
- ***1983*** - TCP/IP replaces ARPANET at U.S. Dept. of Defense, effective birth of Internet
  - First model for net management - **HEMS** - High-Level Entity Management System (*RFCs 1021,1022,1024,1076*)
- ***1987*** - ISO OSI proposes **CMIP** - Common Management Information Protocol, and **CMOT** (CMIP over TCP) for the actual network management protocol for use on the internet
- ***Nov. 1987*** - **SGMP** - Simple Gateway Monitoring protocol (*RFC 1028*)
- ***1989*** - Marshall T. Rose heads up **SNMP** working group to create a common network management framework to be used by both **SGMP** and **CMOT** to allow for transition to **CMOT**
- ***Aug. 1989*** - "**Internet-standard Network Management Framework**" defined (*RFCs 1065, 1066, 1067*)
- ***Apr. 1989*** - **SNMP** promoted to ***recommended*** status as the de facto TCP/IP network management framework (*RFC 1098*)
- ***June 1989*** - IAB committee decides to let **SNMP** and **CMOT** develop separately
- ***May 1990*** - IAB promotes SNMP to a **standard protocol with a recommended status** (*RFC 1157*)
- ***Mar. 1991*** - format of MIBs and traps defined (*RFCs 1212, 1215*)
  - TCP/IP MIB definition revised to create **SNMPv1** (RFC 1213)

SNMP & OSI model

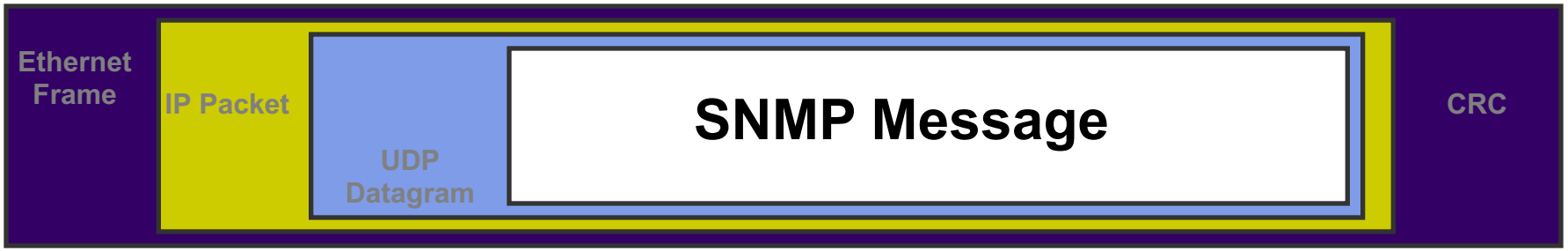


7	Application Layer	Management and Agent APIs
		SNMP
6	Presentation Layer	ASN.1 and BER
5	Session Layer	RPC and NetBIOS
4	Transport Layer	TCP and UDP
3	Network Layer	IP and IPX
2	Data Link Layer	Ethernet, Token Ring, FDDI
1	Physical Layer	

Port & UDP



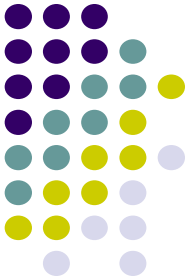
•SNMP uses User Datagram Protocol (UDP) as the transport mechanism for SNMP messages



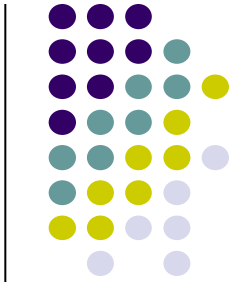
•Like FTP, SNMP uses two well-known ports to operate:

- **UDP Port 161** - SNMP Messages
- **UDP Port 162** - SNMP Trap Messages

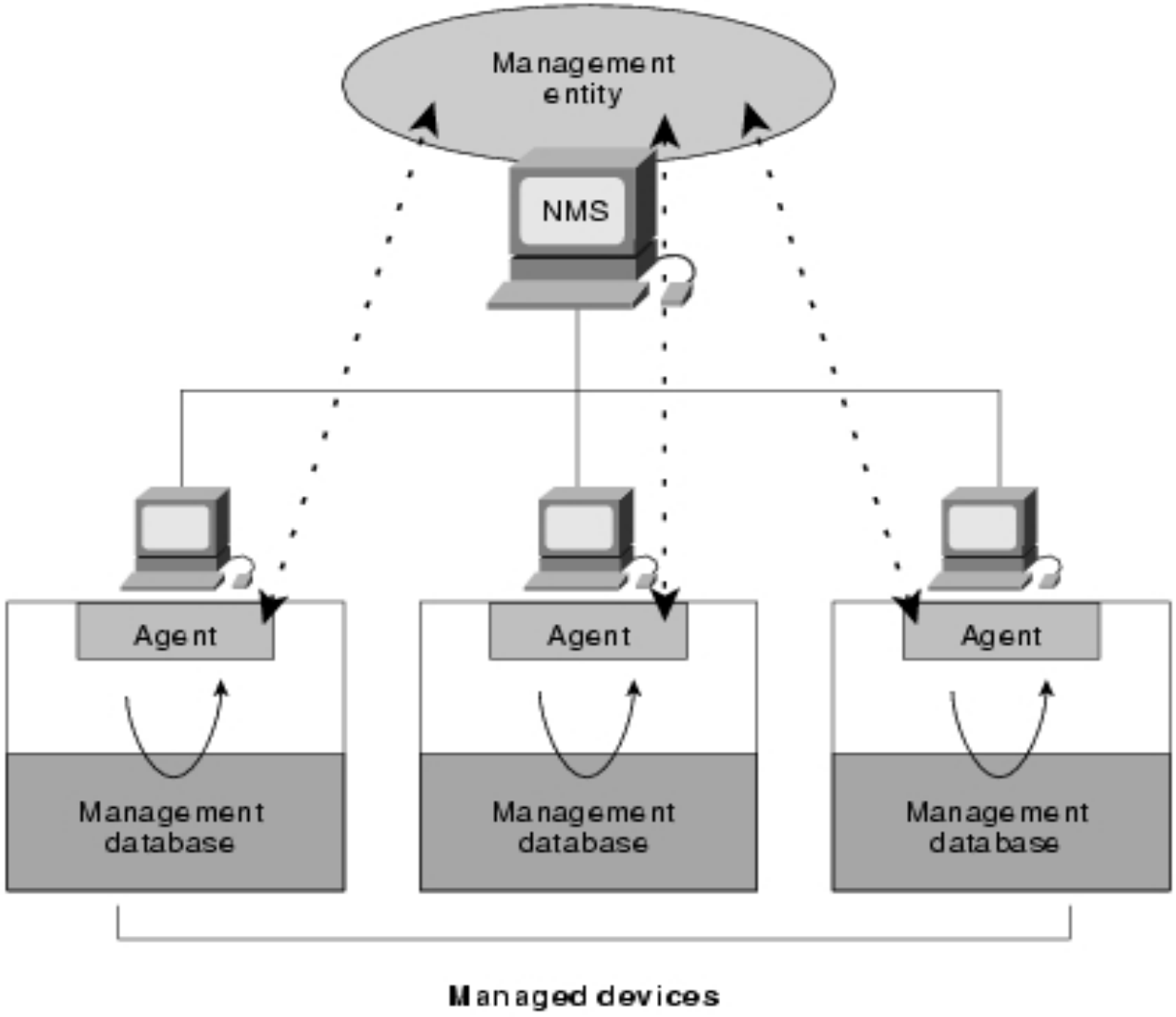
SNMP Components



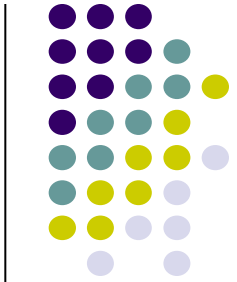
- An SNMP-managed network consists of three key components:
  - ***managed devices***,
  - ***agents***, and
  - ***network-management systems*** (NMSs).



- A *managed device* is a network node that contains an SNMP agent and that resides on a managed network. Managed devices collect and store management information and make this information available to NMSs using SNMP. Managed devices, sometimes called network elements, can be routers and access servers, switches and bridges, hubs, computer hosts, or printers.
- An *agent* is a network-management software module that resides in a managed device. An agent has local knowledge of management information and translates that information into a form compatible with SNMP.
- An *NMS* executes applications that monitor and control managed devices. NMSs provide the bulk of the processing and memory resources required for network management. One or more NMSs must exist on any managed network.

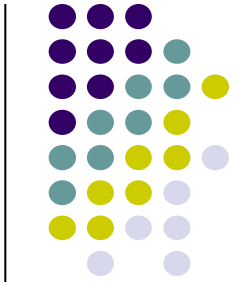

Management entity
NMS
Agent
Agent
Agent
Management database
Management database
Management database
Managed devices

Basic Command



- Managed devices are monitored and controlled using four basic SNMP commands: **read**, **write**, **trap**, and traversal operations.
  - The **read** command is used by an NMS to monitor managed devices. The NMS examines different variables that are maintained by managed devices.
  - The **write** command is used by an NMS to control managed devices. The NMS changes the values of variables stored within managed devices.
  - The **trap** command is used by managed devices to asynchronously report events to the NMS. When certain types of events occur, a managed device sends a trap to the NMS.
  - Traversal operations are used by the NMS to determine which variables a managed device supports and to sequentially gather information in variable tables, such as a routing table.

Language of SNMP



- Structure of Management Information (SMI)

  specifies the format used for defining managed objects that are accessed via the SNMP protocol

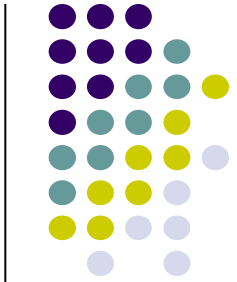
- Abstract Syntax Notation One (ASN.1)

  used to define the format of SNMP messages and managed objects (MIB modules) using an unambiguous data description format

- Basic Encoding Rules (BER)

  used to encode the SNMP messages into a format suitable for transmission across a network

Abstract Syntax Notation One



ASN.1 is nothing more than a language definition. It is similar to C/C++ and other programming languages.

Syntax examples:

```
-- two dashes is a comment -- The C equivalent is written in the comment
MostSevereAlarm ::= INTEGER          -- typedef MostSevereAlarm int;
circuitAlarms MostSevereAlarm ::= 3   -- MostSevereAlarm circuitAlarms = 3;
MostSevereAlarm ::= INTEGER (1..5)  -- specify a valid range
ErrorCounts ::= SEQUENCE {
   circuitID              OCTET STRING,
   erroredSeconds         INTEGER,
   unavailableSeconds  INTEGER
}   -- data structures are defined using the SEQUENCE keyword
```

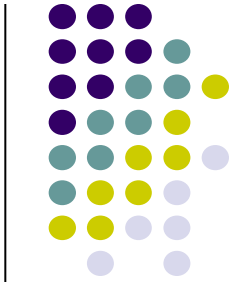
Simple Data Types

RED items defined by ASN.1

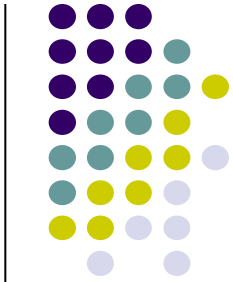
Blue items defined by RFC 1155

- INTEGER -- signed 32-bit integer
- OCTET STRING
- OBJECT IDENTIFIER (OID)
- NULL -- not actually data type, but data value
- IpAddress -- OCTET STRING of size 4, in network byte order (B.E.)
- Counter -- unsigned 32-bit integer (rolls over)
- Gauge -- unsigned 32-bit integer (will top out and stay there)
- TimeTicks -- unsigned 32-bit integer (rolls over after 497 days)
- Opaque -- used to create new data types not in SNMPv1
- DateAndTime, DisplayString, MacAddress, PhysAddress, TimeInterval, TimeStamp, TruthValue, VariablePointer -- textual conventions used as types

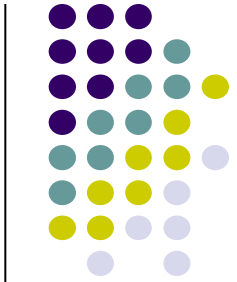
MIB



- *Management Information Base (MIB)* is a collection of information that is organized hierarchically. MIBs are accessed using a network-management protocol such as SNMP. They are comprised of managed objects and are identified by object identifiers.



- Two types of managed objects exist: scalar and tabular
  - *Scalar objects* define a single object instance.
  - *Tabular objects* define multiple related object instances that are grouped in MIB tables.



Always defined and referenced within the context of a MIB

A typical MIB variable definition:

```
sysContact OBJECT-TYPE          -- OBJECT-TYPE is a macro
    SYNTAX      DisplayString (SIZE (0..255))
    ACCESS      read-write      -- or read-write, write-only, not-accessible
    STATUS      mandatory       -- or optional, deprecated, obsolete
    DESCRIPTION
        "CEPN1331 Computer Network"
    ::= { system 4 }
```

MIB – Management Information Base

- MIB Breakdown…

- OBJECT-TYPE
  - String that describes the MIB object.
  - Object IDentifier (OID).
- SYNTAX
  - Defines what kind of info is stored in the MIB object.
- ACCESS
  - READ-ONLY, READ-WRITE.
- STATUS
  - State of object in regards the SNMP community.
- DESCRIPTION
  - Reason why the MIB object exists.

Standard MIB Object:

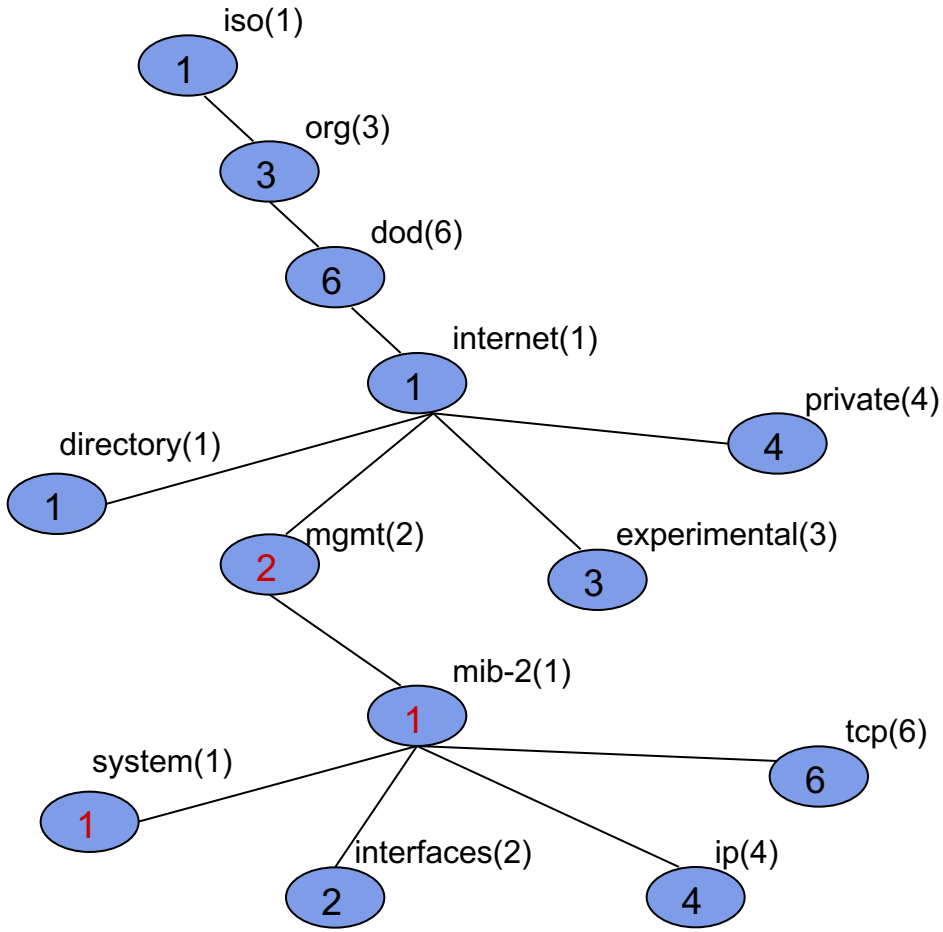
sysUpTime **OBJECT-TYPE**
SYNTAX Time-Ticks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Time since the network management portion of the system was last re-initialised.
::= {system 3}

MIB – Management Information Base

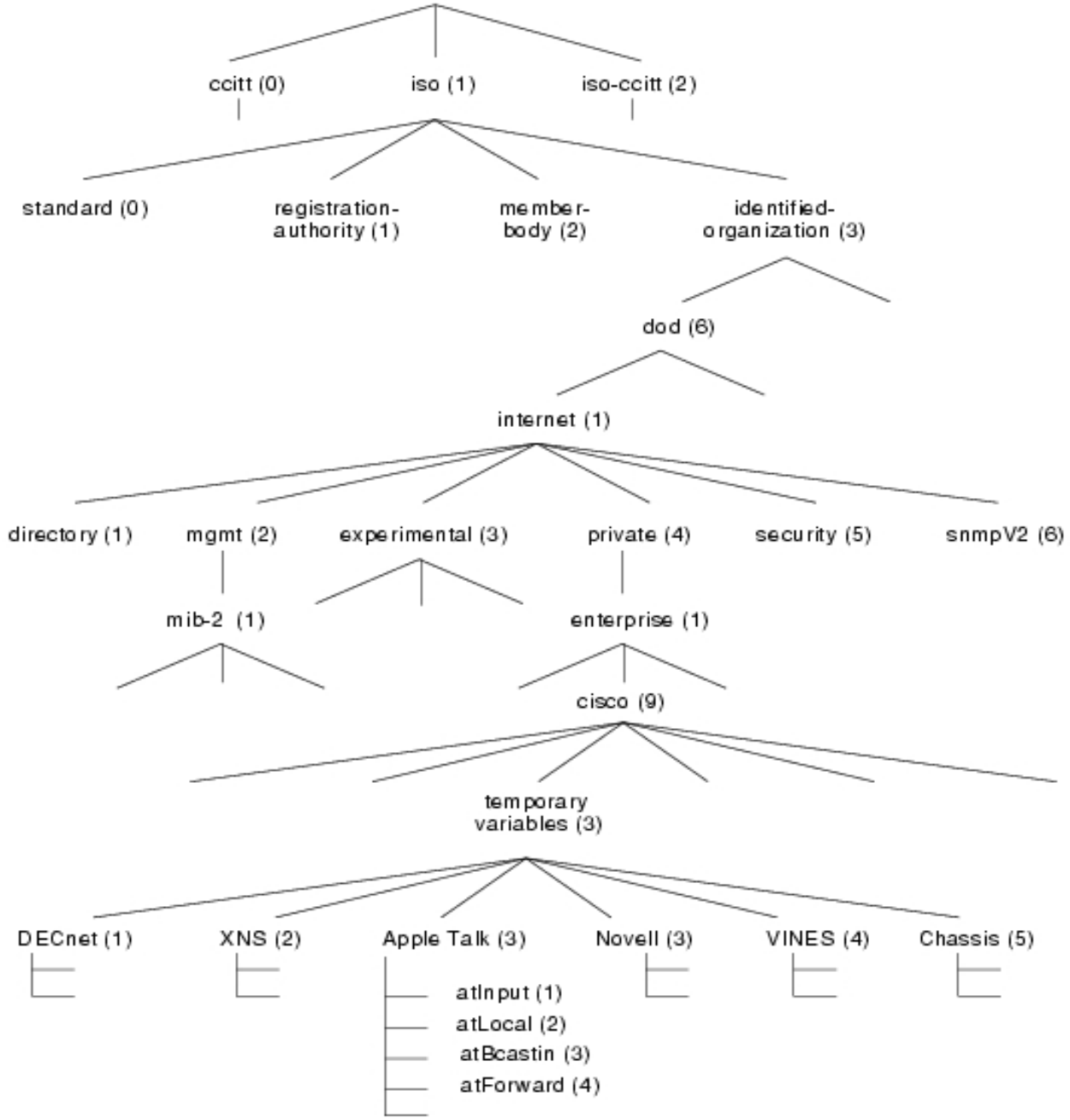
- Object IDentifier (OID)

- Example .1.3.6.1.2.1.1

- iso(1) org(3) dod(6) internet(1)
  mgmt(2)
  mib-2(1)
  system(1)



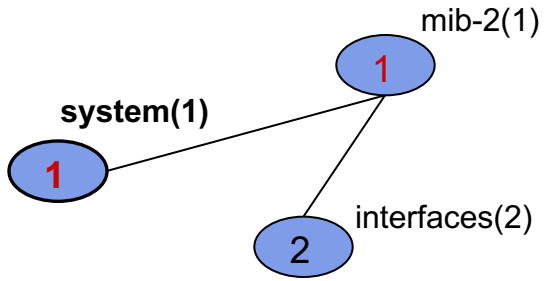
Note:
- .1.3.6.1 ~100% present.
- mgmt and private most common.
- MIB-2 successor to original MIB.
- STATUS ‘mandatory’, All or nothing in group


ccitt (0)
iso (1)
iso-ccitt (2)
standard (0)
registration-authority (1)
member-body (2)
identified-organization (3)
dod (6)
internet (1)
directory (1)
mgmt (2)
experimental (3)
private (4)
security (5)
snmpV2 (6)
mib-2 (1)
enterprise (1)
cisco (9)
temporary variables (3)
DECnet (1)
XNS (2)
Apple Talk (3)
Novell (3)
VINES (4)
Chassis (5)
atInput (1)
atLocal (2)
atBcastin (3)
atForward (4)

MIB – Management Information Base

- system(1) group

- Contains objects that describe some basic information on an entity.
- An entity can be the agent itself or the network object that the agent is on.

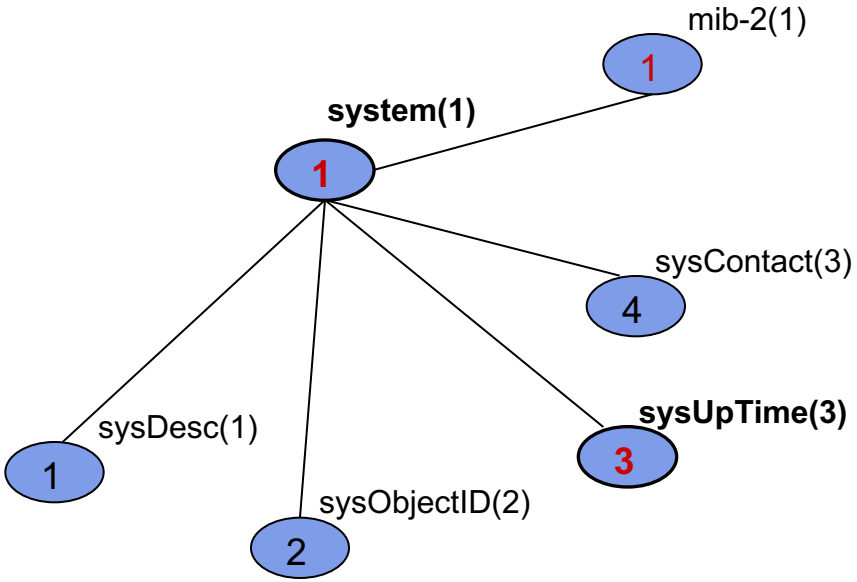


- system(1) group objects

- **sysDescr(1)** → Description of the entity.
- **sysObjectID(2)** → Vendor defined OID string.
- **sysUpTime(3)** → Time since net-mgt was last re-initialised.
- **sysContact(4)** → Name of person responsible for the entity.

MIB – Management Information Base

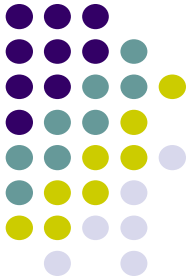
MIB - tree view



MIB - syntax view

sysUpTime **OBJECT-TYPE**
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The time (in hundredths of a second) since the network management portion of the system was last re-initialized."
::= {system 3}

MIB – Management Information Base

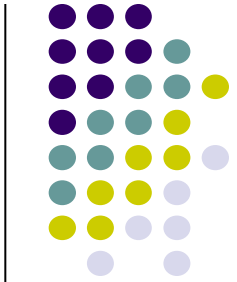


- SNMP Instances
- Each MIB object can have an instance.
  - A MIB for a router's (entity) interface information…

iso(1) org(3) dod(6) internet(1) mgmt(2) mib-2(1) interfaces(2) **ifTable(2) ifEntry(1)** ifType(3)

  - Require one ifType value per interface (e.g. 3)
  - One MIB object definition can represent multiple instances through Tables, Entries, and Indexes.

MIB – Management Information Base

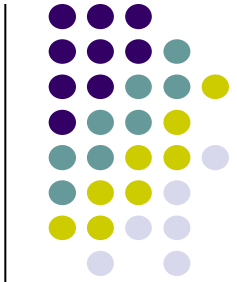


- Tables, Entries, and Indexes.
- Imagine tables as spreadsheets…
  - Three interface types require 3 rows (index no.s)
  - Each column represents a MIB object, as defined by the entry node.

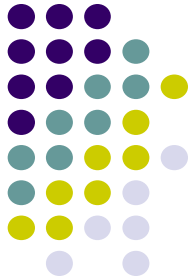
ENTRY + INDEX = INSTANCE

	ifType(3)	**ifMtu(4)**	**Etc…**
Index #1	ifType.1[**6**]	ifMtu.1	
Index #2	ifType.2:[**9**]	ifMtu.2	
Index #3	ifType.3:[**15**]	ifMtu.3	

Simple Network Management Protocol



- Retrieval protocol for MIB.
- Can retrieve by
  - CLI (snmpwalk),
  - GUI (MIB Browser), or
  - Larger applications (Sun Net Manager) called Network Management Software (NMS).
- NMS collection of smaller applications to manage network with illustrations, graphs, etc.
- NMS run on Network Management Stations (also NMS), which can run several different NMS software applications.

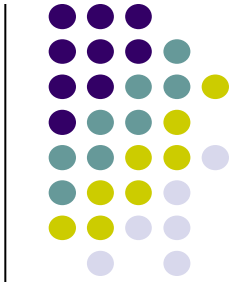


SNMP Commands

- SNMP has 5 different functions referred to as Protocol Data Units (PDU's), which are:

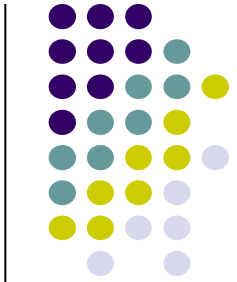
  (1) GetRequest, aka Get
  (2) GetNextRequest, aka GetNext
  (3) GetResponse, aka Response
  (4) SetRequest, aka Set
  (5) Trap

SNMP Commands [Get]



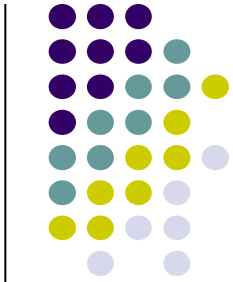
- GetRequest [Get]
  - Most common PDU.
  - Used to ask SNMP agent for value of a particular MIB agent.
  - NMS sends out 1 Get PDU for each instance, which is a unique OID string.
  - What happens if you don’t know how many instances of a MIB object exist?

SNMP Commands [GetNext]



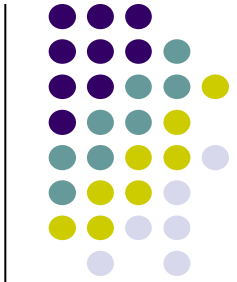
- GetNextRequest [GetNext]
  - NMS application uses GetNext to ‘walk’ down a table within a MIB.
  - Designed to ask for the OID and value of the MIB instance that comes after the one asked for.
  - Once the agent responds the NMS application can increment its count and generate a GetNext.
  - This can continue until the NMS application detects that the OID has changed, i.e. it has reached the end of the table.

SNMP Commands [GetResponse]



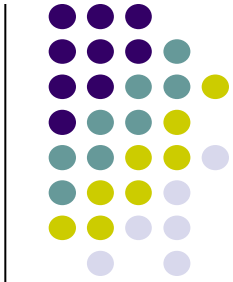
- GetResponse [Response]
  - Simply a response to a Get, GetNext or Set.
  - SNMP agent responds to all requests or commands via this PDU.

SNMP Commands [SetRequest]



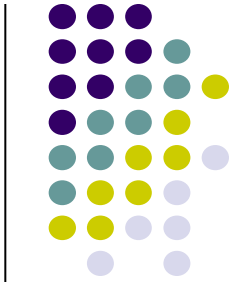
- SetRequest [Set]
  - Issued by an NMS application to change a MIB instance to the variable within the Set PDU.
  - For example, you could issue a
    - GetRequest against a KDEG server asking for sysLocation.0 and may get ‘ORI’ as the response.
    - Then, if the server was moved, you could issue a Set against that KDEG server to change its location to ‘INS’.
  - You must have the correct permissions when using the set PDU.

SNMP Commands [Trap]



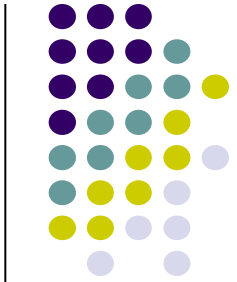
- Trap
  - Asynchronous notification.
  - SNMP agents can be programmed to send a trap when a certain set of circumstances arise.
  - Circumstances can be view as thresholds, i.e. a trap may be sent when the temperature of the core breaches a predefined level.

SNMP Security



- SNMP Community Strings (like passwords)
  - 3 kinds:
    - READ-ONLY: You can send out a Get & GetNext to the SNMP agent, and if the agent is using the same read-only string it will process the request.
    - READ-WRITE: Get, GetNext, and Set. If a MIB object has an ACCESS value of read-write, then a Set PDU can change the value of that object with the correct read-write community string.
    - TRAP: Allows administrators to cluster network entities into communities. Fairly redundant.

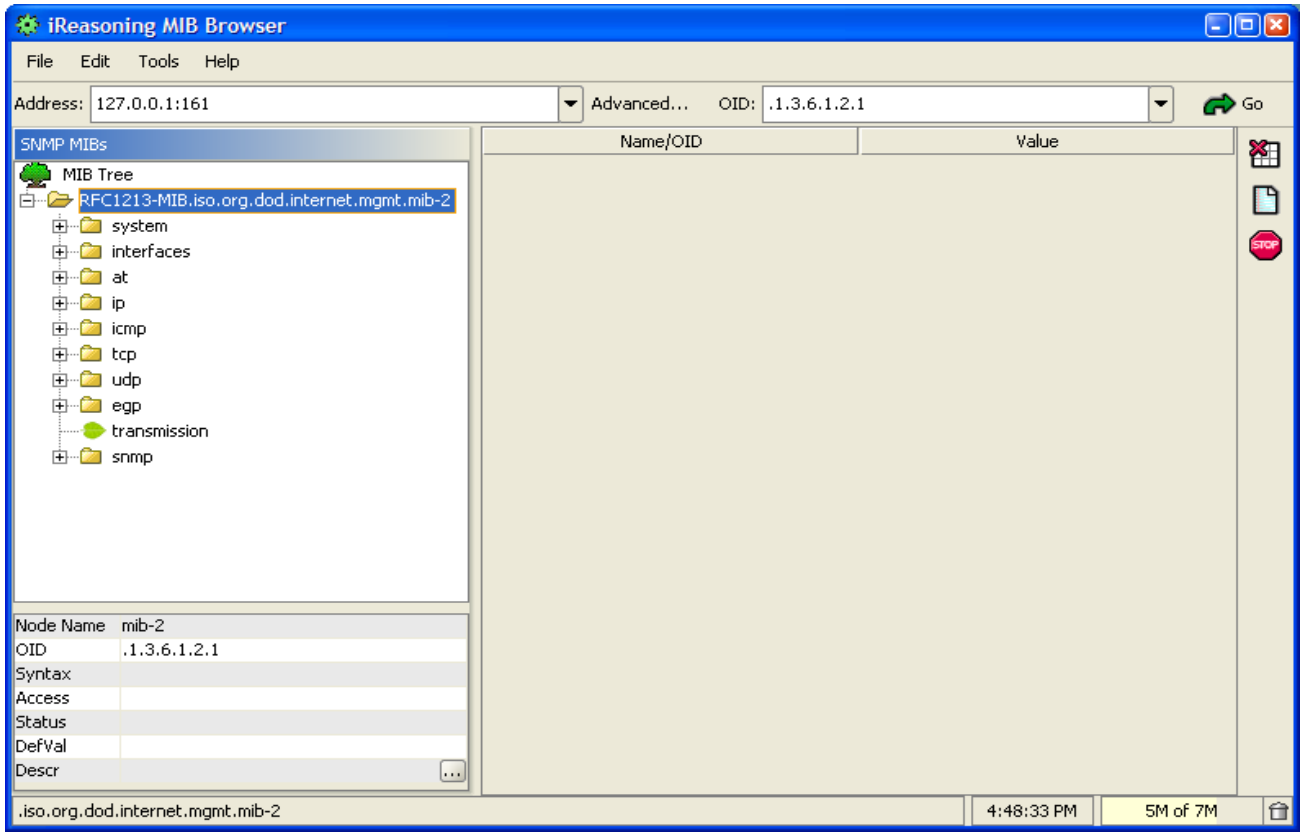
SNMP Tools



- Command Line Interface
  - e.g. ‘snmpwalk’
- Graphical User Interface
  - e.g. iReasoning’s MIB Browser
    - Or via www.ireasoning.com

SNMP – MIB Browser (1)

- Initial set-up… java -Xmx384m -jar "XYZ\lib\browser.jar" (where XYZ = your specific path)



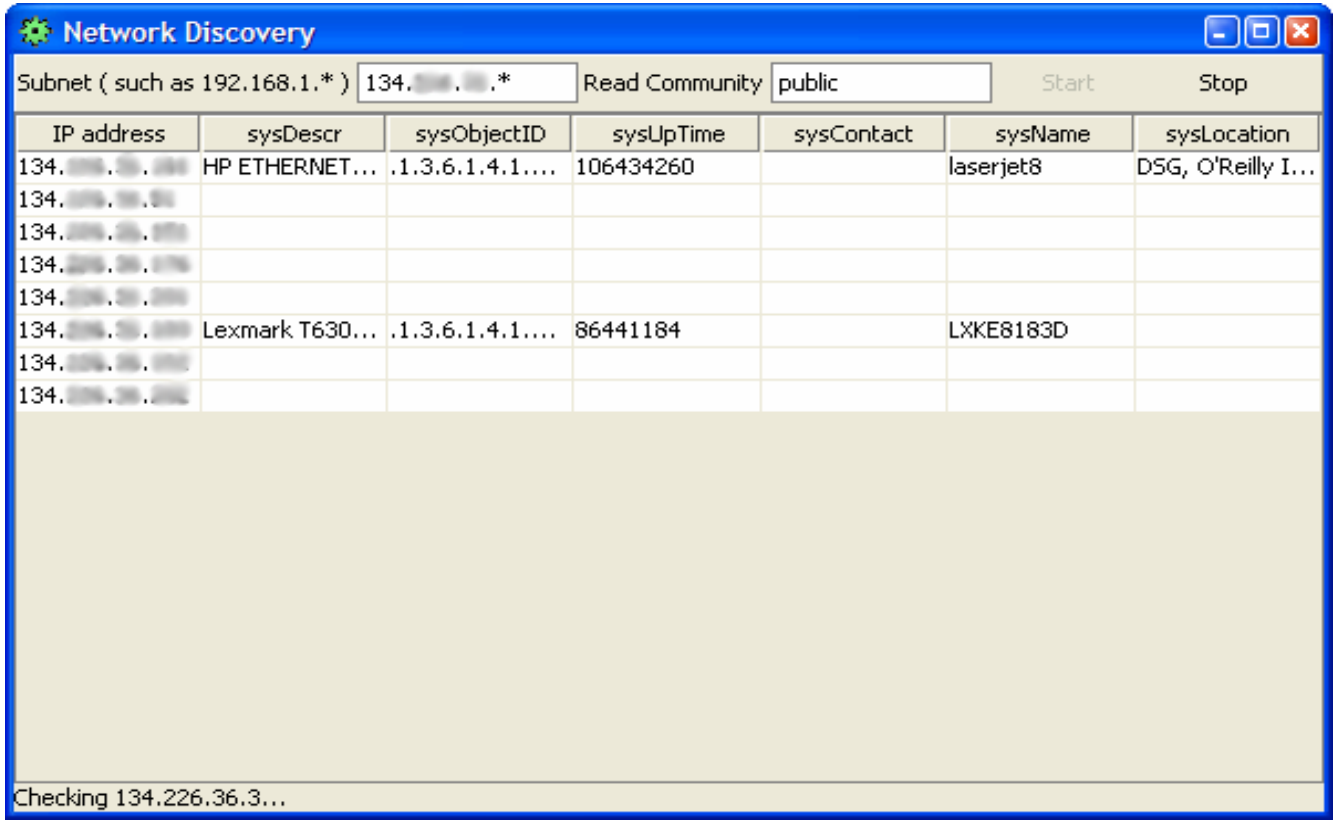
Breakdown…

- LHS is the SNMP MIB structure.

- Lower LHS has details of MIB structure.

- RHS will present MIB values.

SNMP – MIB Browser (2)



Discovery…

- Subnet: 134.XXX.XXX.*

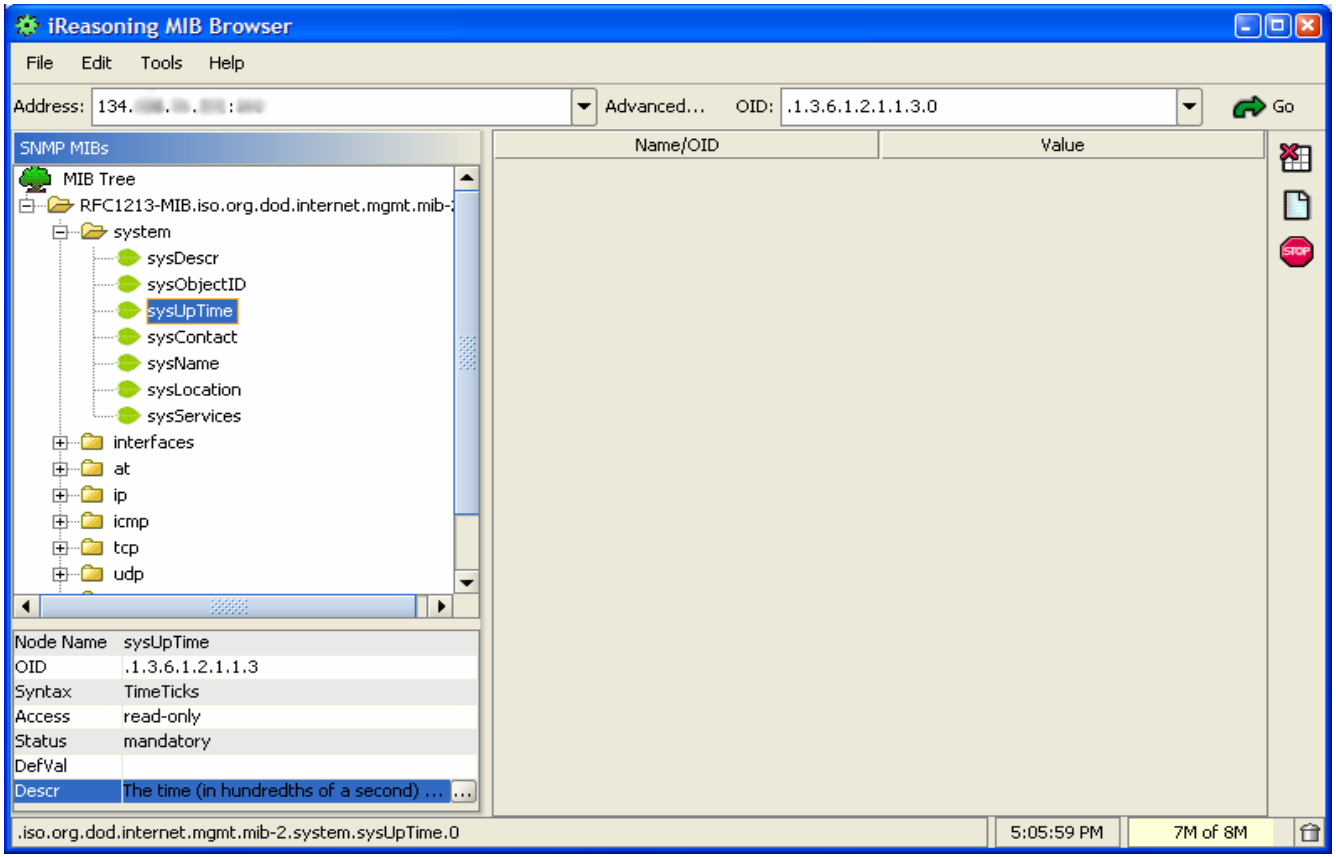
- Read Community: public

→ Start

Note IP Address.

← Stop

SNMP – MIB Browser (3)



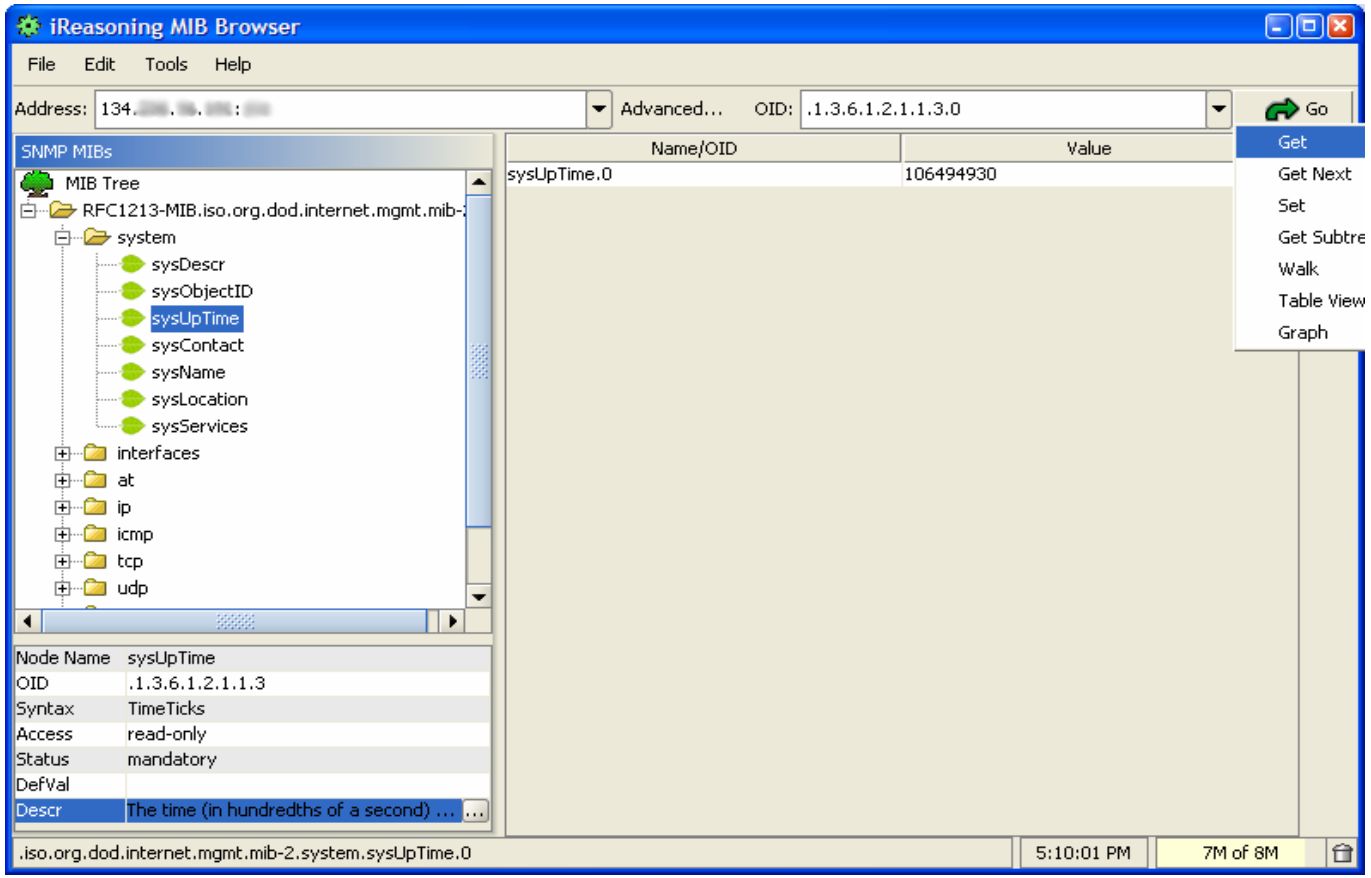
Navigation…

- MIB Tree
  - → System
    - → sysUpTime

-Notice Lower LHS

- Notice OID

SNMP – MIB Browser (4)



SNMP PDU's…

(1) Get

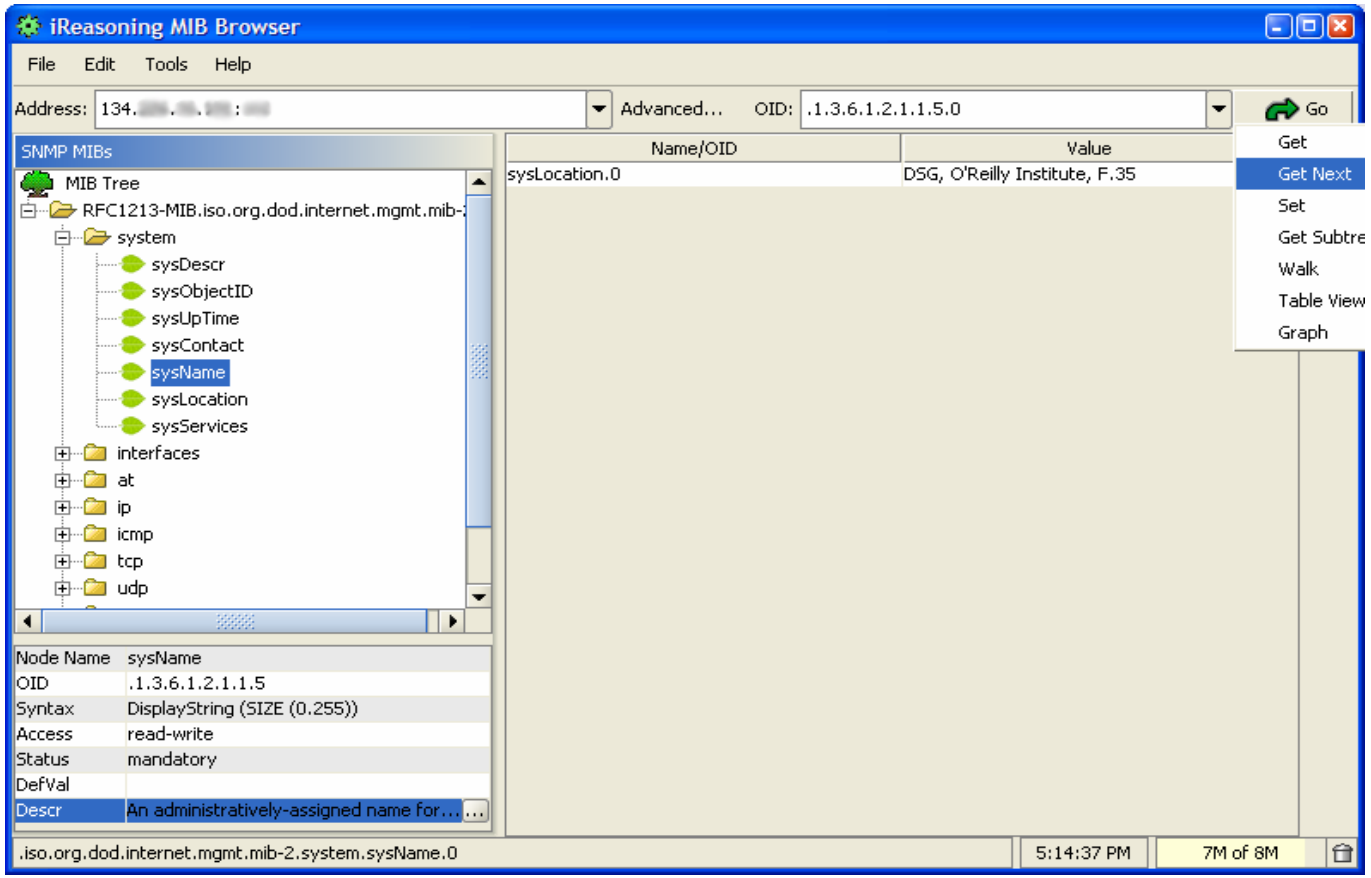
- Select 'Go'

→ 'Get'

- RHS has values.

- OID – Value

SNMP – MIB Browser (5)



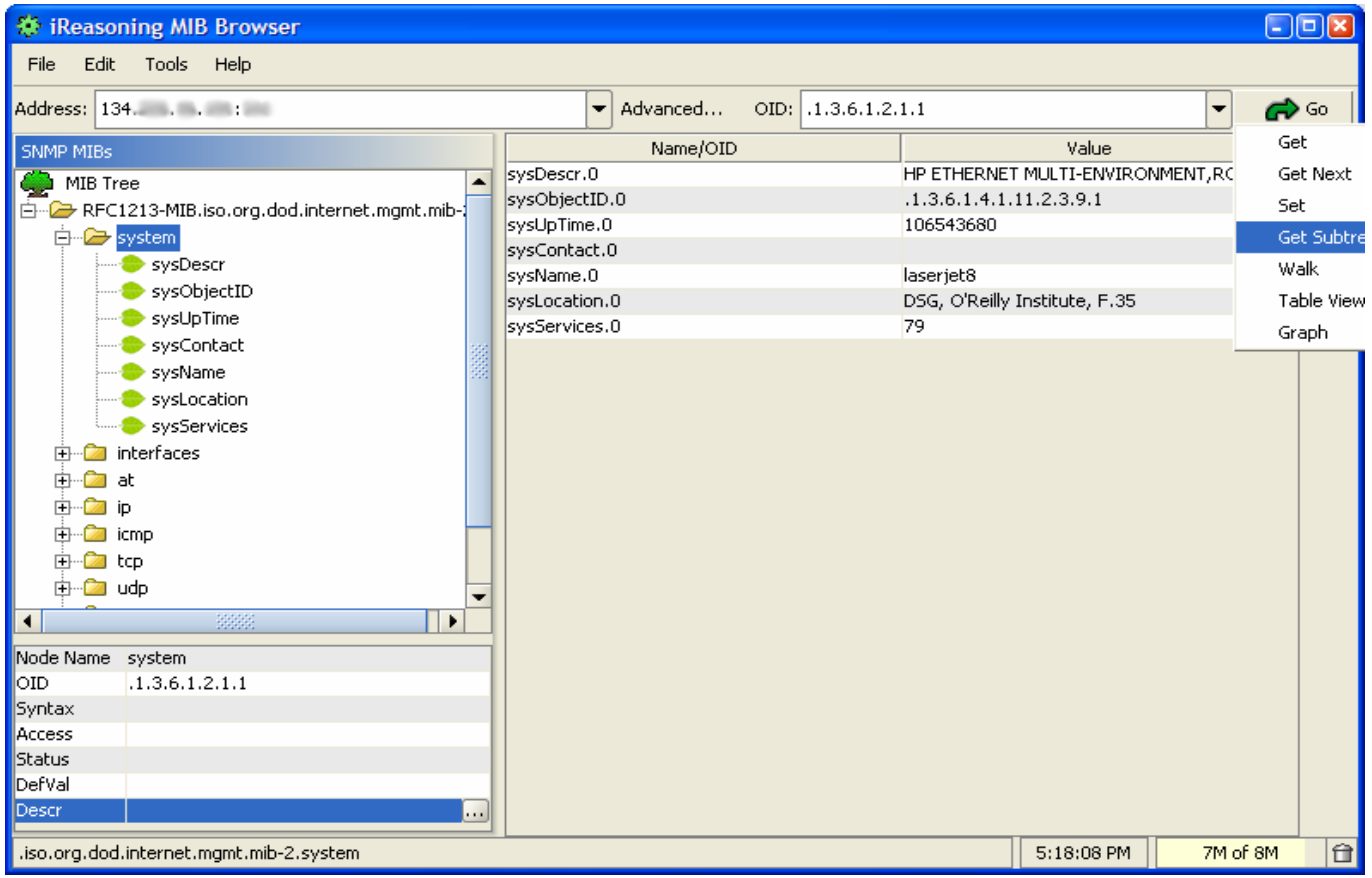
SNMP PDU's…

(2) GetNext

-Selected OID is:
.1.3.6.1.2.1.1.5

-Returned value:
(.1.3.6.1.2.1.1.6)
or
"DSG, O'Reilly Institute,
F.35"

SNMP – MIB Browser (6)



SNMP…

(3) Get SubTree

-Position of MIB:

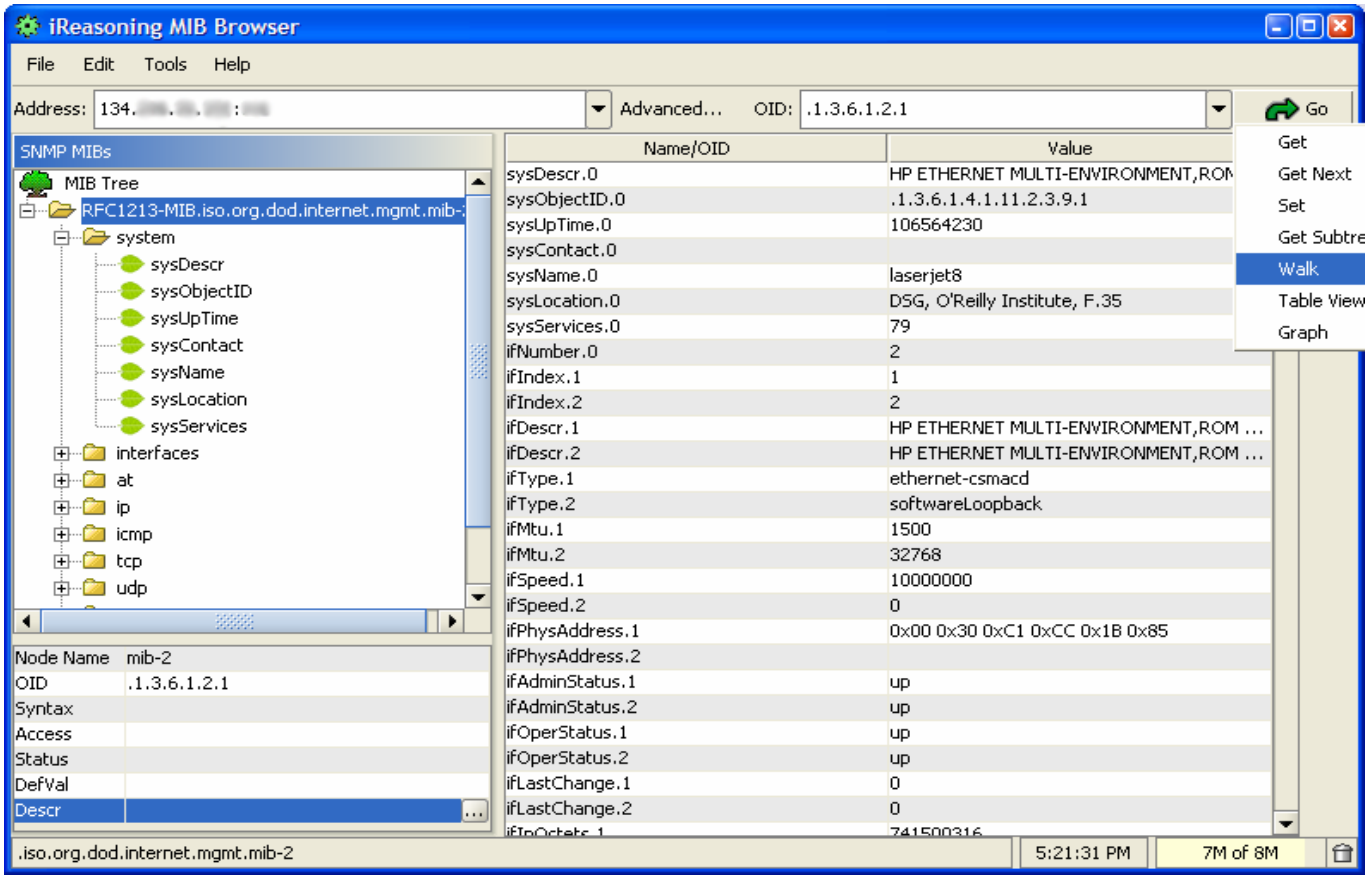
.1.3.6.1.2.1.1

(a.k.a. system)

-RHS values:

Returns all values below system.

SNMP – MIB Browser (7)



SNMP…

(4) Walk

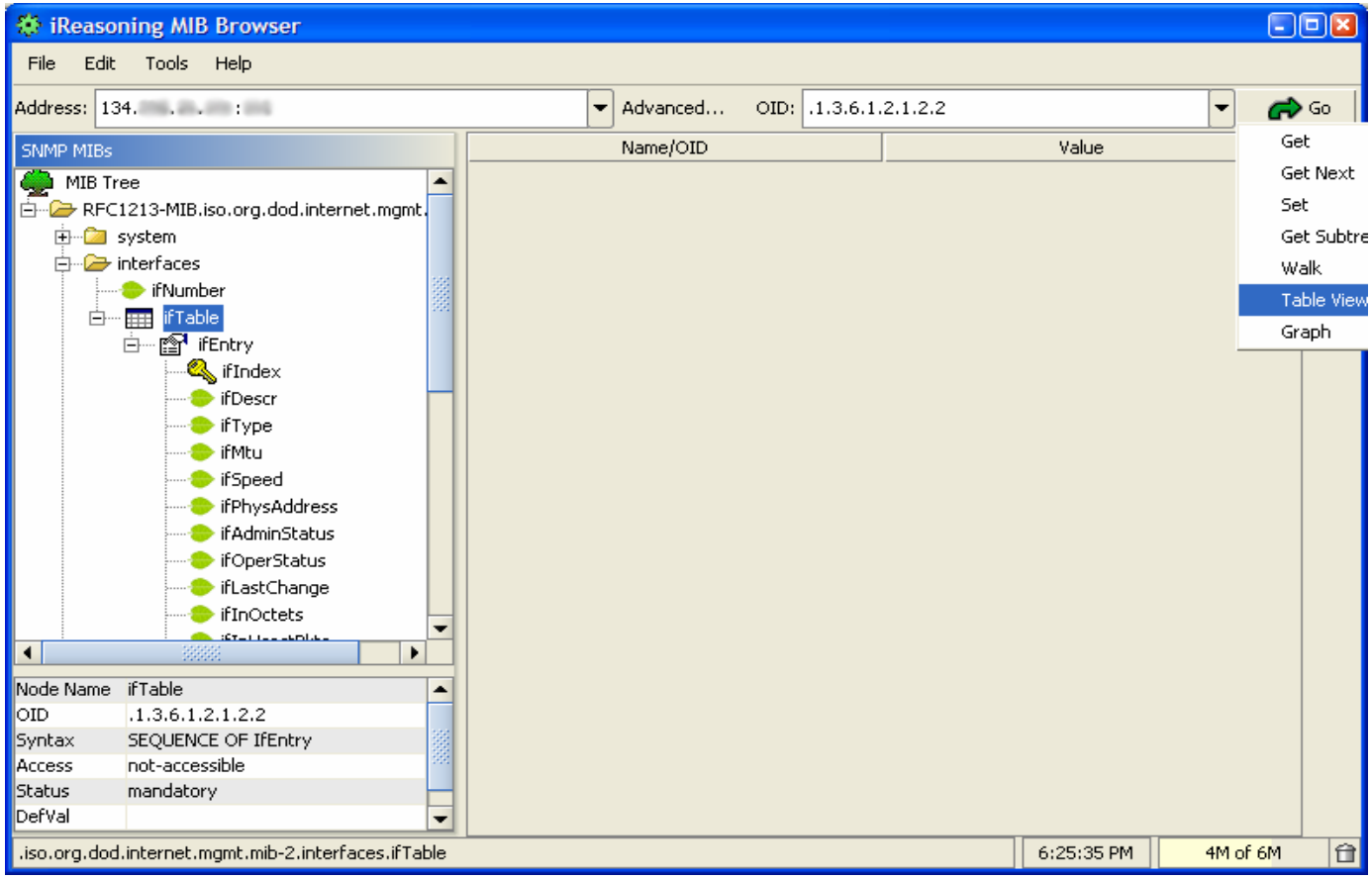
-MIB Location:

.1.3.6.1.2.1

(a.k.a. mib-2)

- Returns *ALL* values under mib-2

SNMP – MIB Browser (8)

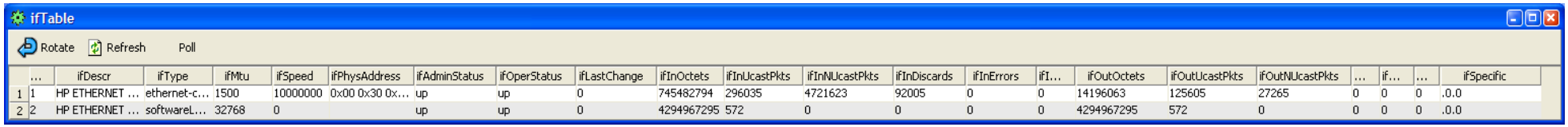


Tables…

- MIB Location:
.1.3.6.1.2.1.2.2
(or interfaces)

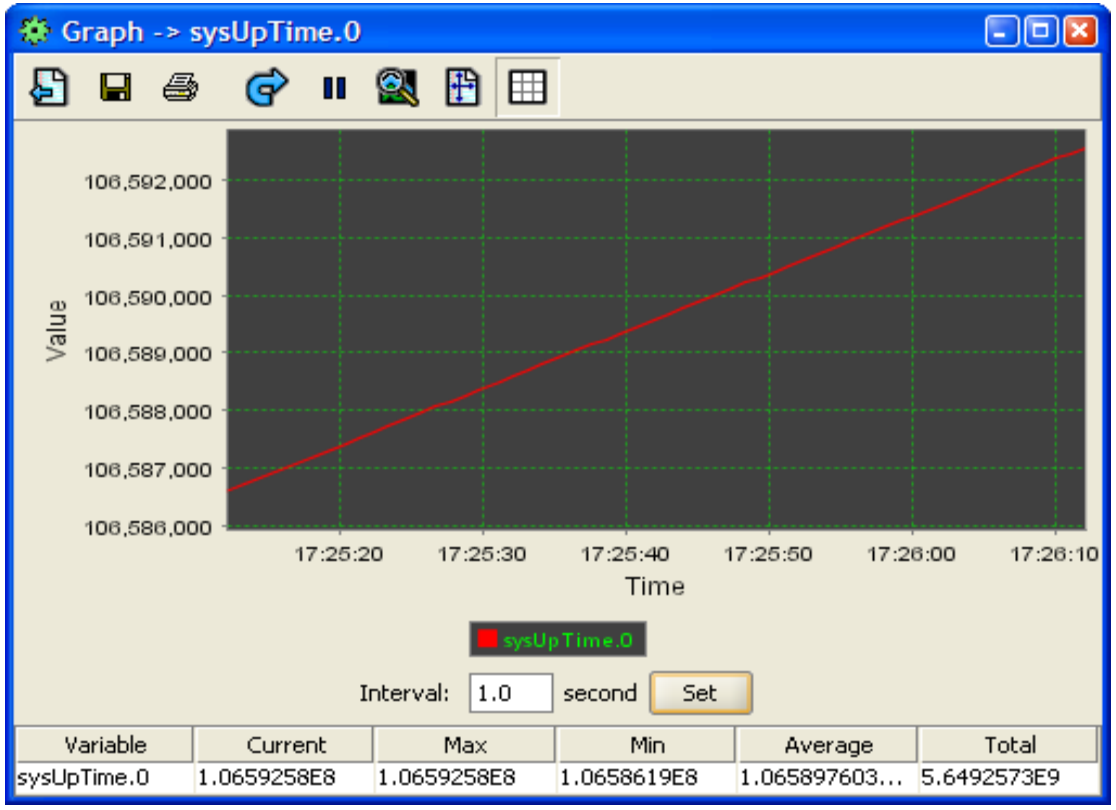
- Select ifTable,
→ Go, then Table View.

- Refresh/Poll



	...	ifDescr	ifType	ifMtu	ifSpeed	ifPhysAddress	ifAdminStatus	ifOperStatus	ifLastChange	ifInOctets	ifInUcastPkts	ifInNUcastPkts	ifInDiscards	ifInErrors	ifI...	ifOutOctets	ifOutUcastPkts	ifOutNUcastPkts	...	if...	...	ifSpecific
1	1	HP ETHERNET ...	ethernet-c...	1500	10000000	0x00 0x30 0x...	up	up	0	745482794	296035	4721623	92005	0	0	14196063	125605	27265	0	0	0	.0.0
2	2	HP ETHERNET ...	softwareL...	32768	0		up	up	0	4294967295	572	0	0	0	0	4294967295	572	0	0	0	0	.0.0

SNMP – MIB Browser (9)



SNMP…

- Graph

- Select a value from the RHS, say sysUpTime
- Highlight and select ‘Go’, then ‘Graph’.
- Interval = 1s → set.