Network monitoring

SNMP and Nagios

Network Management

- introduction to network management
  - motivation
  - major components
- Internet network management framework
  - MIB: management information base
  - SMI: data definition language
  - SNMP: protocol for network management
  - security and administration

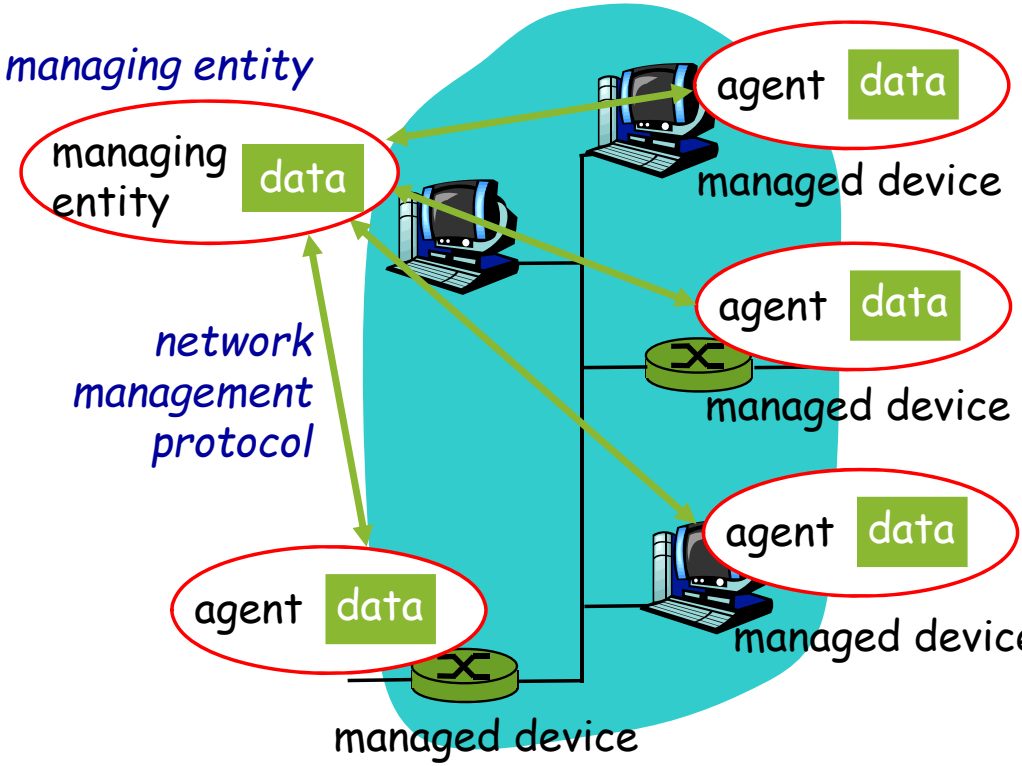
What is network management?

- autonomous systems (aka "network"): 100s or 1000s of interacting hardware/software components
- other complex systems requiring monitoring, control:
  - jet airplane
  - nuclear power plant
  - others?

"Network management includes the deployment, integration and coordination of the hardware, software, and human elements to monitor, test, poll, configure, analyze, evaluate, and control the network and element resources to meet the real-time, operational performance, and Quality of Service requirements at a reasonable cost."

Infrastructure for network management

definitions:



managed devices contain *managed objects* whose data is gathered into a *Management Information Base (MIB)*

Network Management standards

OSI CMIP

- Common Management Information Protocol
- designed 1980's: *the* unifying net management standard
- too slowly standardized

SNMP: Simple Network Management Protocol

- Internet roots (SGMP)
- started simple
- deployed, adopted rapidly
- growth: size, complexity
- currently: SNMP V3
- *de facto* network management standard

SNMP overview: 4 key parts

- Management information base (MIB):
  - distributed information store of network management data
- Structure of Management Information (SMI):
  - data definition language for MIB objects
- SNMP protocol
  - convey manager<->managed object info, commands
- security, administration capabilities
  - major addition in SNMPv3

SMI: data definition language

Purpose: syntax, semantics of management data well-defined, unambiguous

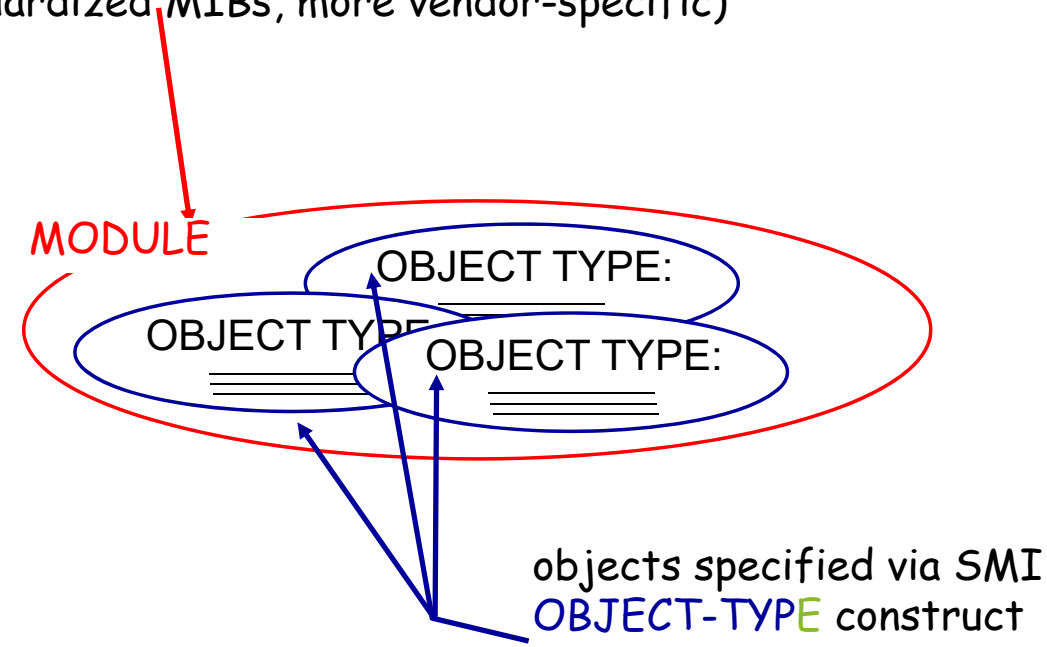
- base data types:
  - straightforward, boring
- OBJECT-TYPE
  - data type, status, semantics of managed object
- MODULE-IDENTITY
  - groups related objects into MIB module

Basic Data Types

INTEGER
Integer32
Unsigned32
OCTET STRING
OBJECT IDENTIFIED
IPaddress
Counter32
Counter64
Guage32
Time Ticks
Opaque

SNMP MIB

MIB module specified via SMI
MODULE-IDENTITY
(100 standardized MIBs, more vendor-specific)



SMI: Object, module examples

OBJECT-TYPE: ipInDelivers

```
ipInDelivers OBJECT TYPE
  SYNTAX        Counter32
  MAX-ACCESS    read-only
  STATUS    current
  DESCRIPTION
     "The total number of input
      datagrams successfully
      delivered to IP user-
      protocols (including ICMP)"
::= { ip   9}
```

MODULE-IDENTITY: ipMIB

```
ipMIB MODULE-IDENTITY
  LAST-UPDATED "941101000Z"
  ORGANZATION "IETF SNPv2
           Working Group"
  CONTACT-INFO
    " Keith McCloghrie
      ……"
  DESCRIPTION
    "The MIB module for managing IP
    and ICMP implementations, but
    excluding their management of
    IP routes."
  REVISION "019331000Z"
 ……
::= {mib-2 48}
```

MIB example: UDP module

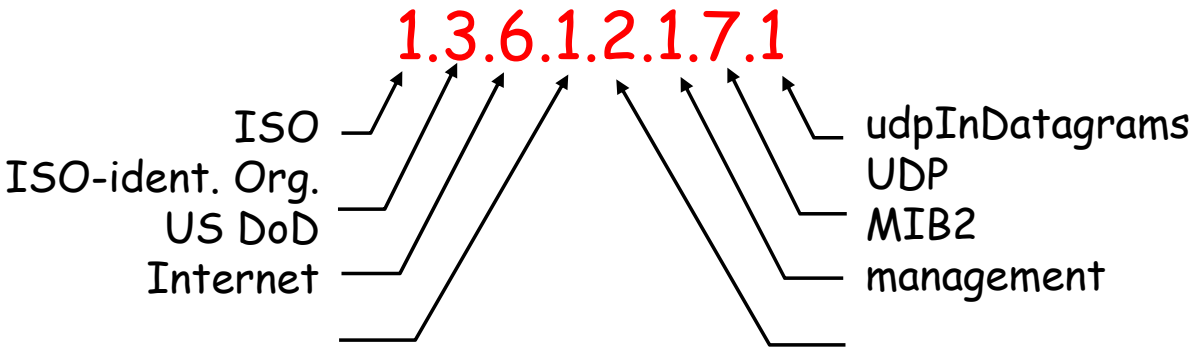
Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPInDatagrams	Counter32	total # datagrams delivered at this node
1.3.6.1.2.1.7.2	UDPNoPorts	Counter32	# underliverable datagrams no app at portl
1.3.6.1.2.1.7.3	UDInErrors	Counter32	# undeliverable datagrams all other reasons
1.3.6.1.2.1.7.4	UDPOutDatagrams	Counter32	# datagrams sent
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	one entry for each port in use by app, gives port # and IP address

SNMP Naming

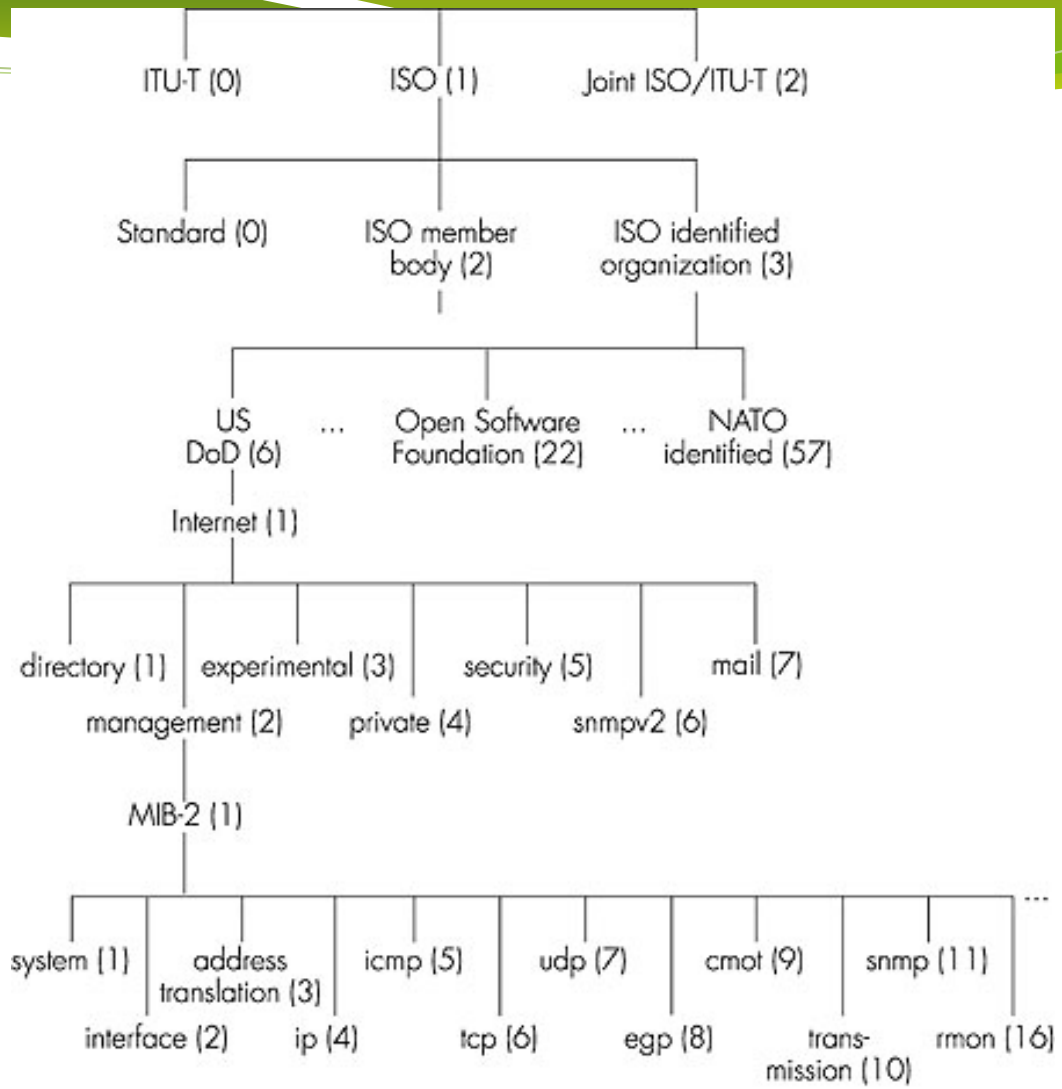
question: how to name every possible standard object (protocol, data, more..) in every possible network standard??

answer: *ISO Object Identifier tree:*

- hierarchical naming of all objects
- each branchpoint has name, number



OSI Object Identifier Tree

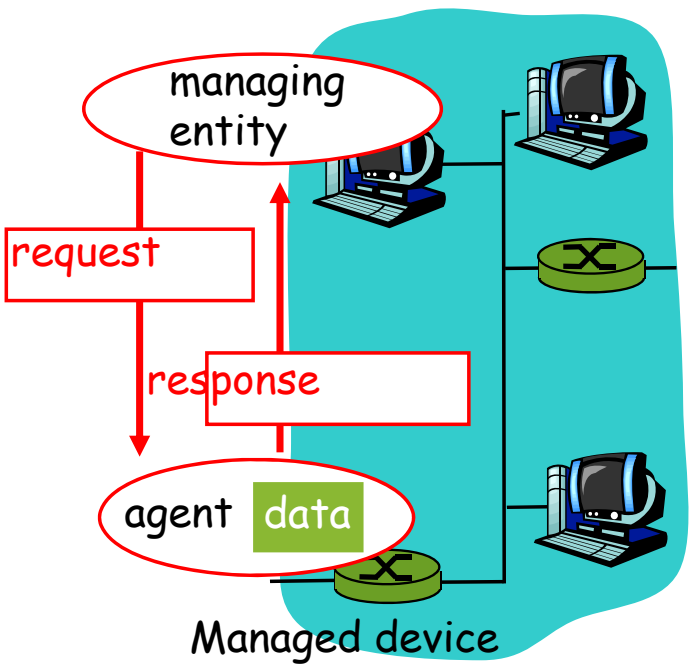


```
$ snmpwalk -v1 -c public 10.10.1.224 .1.3.6.1.4.1.318
SNMPv2-SMI::enterprises.318.1.1.1.1.1.1.0 = STRING: "Silcon DP340E"
SNMPv2-SMI::enterprises.318.1.1.1.1.1.2.0 = STRING: "UPS_IDEN"
SNMPv2-SMI::enterprises.318.1.1.1.1.2.1.0 = STRING: "314.10.D"
```

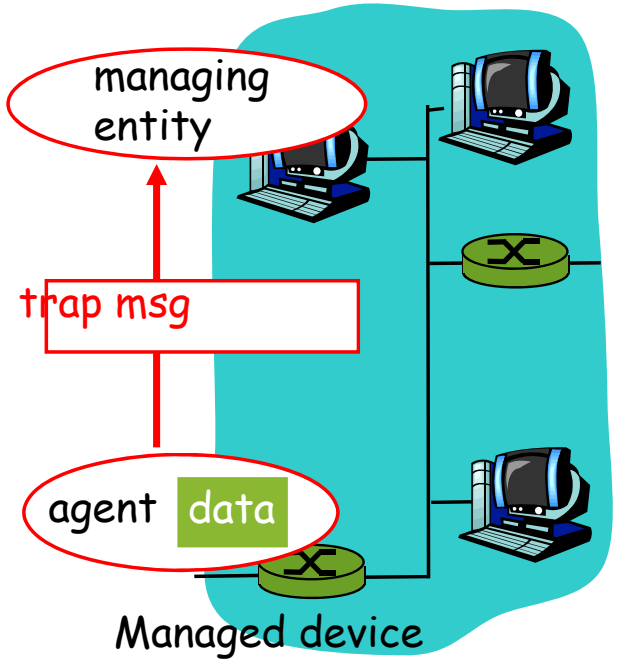
```
$ snmpwalk -v1 -c public -m "./APC-POWERNET.txt" 10.10.1.224 apc
PowerNet-MIB::upsBasicIdentModel.0 = STRING: "Silcon DP340E"
PowerNet-MIB::upsBasicIdentName.0 = STRING: "UPS_IDEN"
PowerNet-MIB::upsAdvIdentFirmwareRevision.0 = STRING: "314.10.D"
```

SNMP protocol

Two ways to convey MIB info, commands:



request/response mode

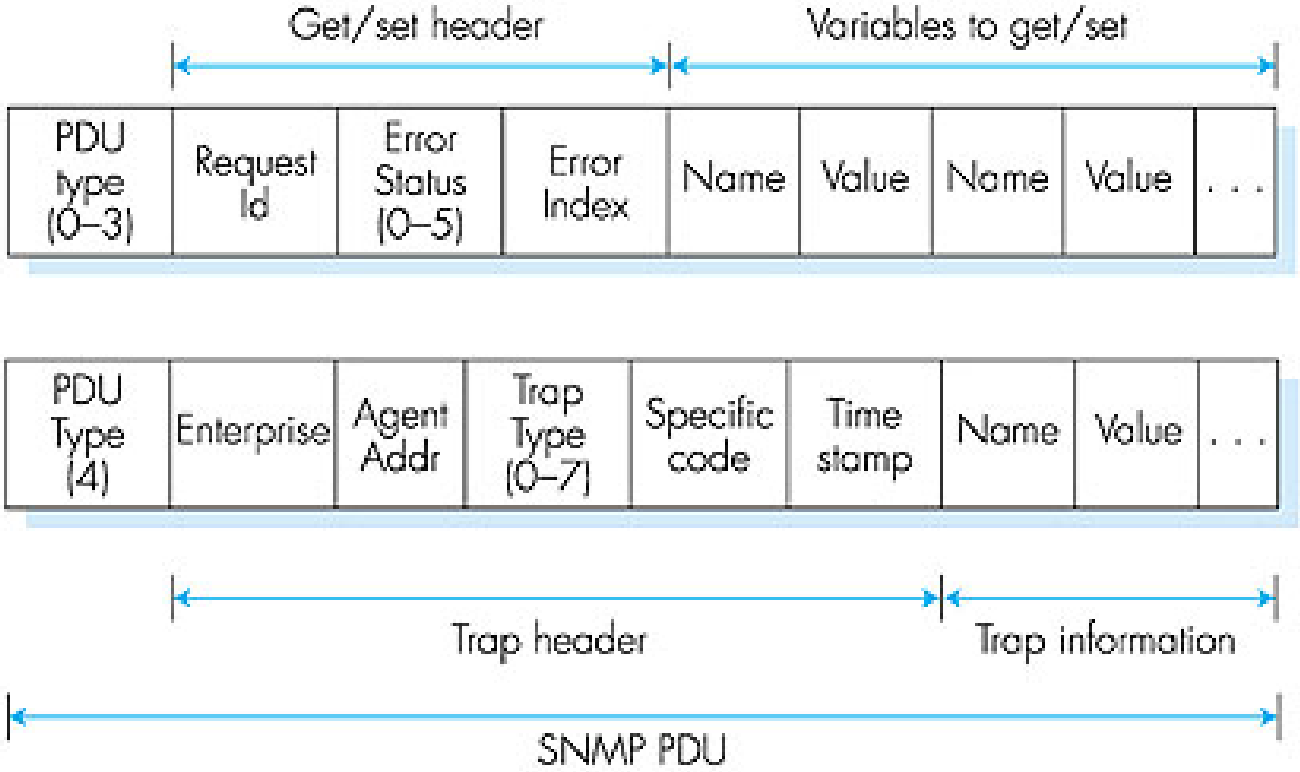


trap mode

SNMP protocol: message types

Message type	Function
GetRequest GetNextRequest GetBulkRequest	Mgr-to-agent: "get me data" (instance,next in list, block)
InformRequest	Mgr-to-Mgr: here's MIB value
SetRequest	Mgr-to-agent: set MIB value
Response	Agent-to-mgr: value, response to Request
Trap	Agent-to-mgr: inform manager of exceptional event

SNMP protocol: message formats



SNMP security and administration

- encryption: DES-encrypt SNMP message
- authentication: compute, send MIC(m,k): compute hash (MIC) over message (m), secret shared key (k)
- protection against playback: use nonce
- view-based access control
  - SNMP entity maintains database of access rights, policies for various users
  - database itself accessible as managed object!

Installing snmp simulator

- `sudo apt-get install snmp snmp-mibs-downloader`
- `sudo apt-get install snmpsim`
- `sudo mkdir /usr/snmpsim/`
- `sudo mkdir /usr/snmpsim/data`

Configuring snmpsimd

- `sudo cp -r /usr/share/doc/snmpsim/examples/data/* /usr/snmpsim/data/`
- `sudo mkdir /var/log/snmpsim/`
- `sudo snmpsimd --agent-udpv4-endpoint=127.0.0.1:161 --process-user=nobody --process-group=nogroup --logging-method=file:/var/log/snmpsim/snmpsimd.log`

In a different terminal enter:

- `snmpwalk -v2c -c recorded/linksys-system 127.0.0.1`

Nagios

- Make sure that you have a couple of services installed
  - `sudo apt-get install openssh-server apache2`
- `sudo apt-get install nagios3`
- Select internet
- input the nagios password and confirm it
- Open the browser and navigate to http://ipaddress/nagios3

Exercise 1

- Install a new service on the Ubuntu box (f.x. FTP or similar)
  - `apt-get install vsftpd`

- Add it to the monitoring in the nagios

Exercise 1 walk-through

- Try out the ftp tester plugin for Nagios:

```
/usr/lib/nagios/plugins/check_ftp -H localhost
```

- Then add the service to the Nagios monitoring:

```
sudo nano /etc/nagios3/conf.d/services_nagios2.cfg

define service {
  hostgroup_name                  ssh-servers
  service_description             FTP
  check_command                   check_ftp
  use                                     generic-service
  notification_interval           0
}
```

- Check that the Nagios config doesn't contain errors

```
sudo nagios3 -v /etc/nagios3/nagios.cfg
```

- And restart Nagios

```
sudo service nagios3 start
```

Exercise 2

- Try to add monitoring to the Object identifier 1.3.6.1.2.1.1.8.0 (this find timeticks)
- You can use the check_snmp for that

Exercise 2 walk-through

- Use the Nagios check_snmp plugin to test (1 line)

```
/usr/lib/nagios/plugins/check_snmp -H 127.0.0.1 -o
1.3.6.1.2.1.1.8.0 -P 2c -C recorded/linksys-system
```

- Create a customized command to this snmp OID in the file snmp.cfg

```
sudo nano  /etc/nagios-plugins/config/snmp.cfg

define command{
        command_name    snmp_ticks
        command_line    /usr/lib/nagios/plugins/check_snmp -H
'$HOSTADDRESS$' -C '$ARG1$' -o 1.3.6.1.2.1.1.8.0 -P 2c
        }
```

Exercise 2 walk-through

- Then add the service to the Nagios monitoring:

```
sudo nano /etc/nagios3/conf.d/services_nagios2.cfg

define service {
  hostgroup_name                  ssh-servers
  service_description             TimeTicks
  check_command                   snmp_ticks!recorded/linksys-system
  use                             generic-service
  notification_interval           0 ; set > 0 if you want to be renotified
}
```

- Check that the Nagios config doesn't contain errors

```
sudo nagios3 -v /etc/nagios3/nagios.cfg
```

- And restart Nagios

```
sudo service nagios3 start
```

References

- SNMP: Chapter 2 (Essential SNMP, 2nd edition, O'Reilly)
- Nagios: http://www.aims-conference.org/issnsm-2008/07-nagios.pdf