



Introduction to SNMP & MRTG

bdNOG 6 – Bogra, Bangladesh

Overview

- What is SNMP?
- Polling and querying
- OIDs and MIBs
- Notifications
- SNMPv3

What is SNMP?

- SNMP - Simple Network Management Protocol
 - Structured protocol, structured information
 - For querying network device state and receiving notifications
 - Also can be used to change state
 - Industry standard, hundreds of tools exist that use it
 - Supported on any decent network equipment
 - Transport : UDP ports 161 and 162 (notifications)

Uses for SNMP

- Typical queries
 - Bytes In/Out on an interface, errors
 - CPU load
 - Uptime
 - Temperature or other vendor specific OIDs
- For hosts (servers or workstations)
 - Disk space
 - Installed software
 - Running processes
- Windows and UNIX have SNMP agents

SNMP Versions

- v1 (1988) Original specification
 - Historic
- v2 (1996) Failed Standard
 - Security+new data types+new operators
 - 64-bit counters, get-bulk, v2 notifications
 - View-based access control model (VACM) introduced
 - Historic, no current implementations left
- v2c (1996) De facto standard
 - v2 data types and operators
 - v1 security (community string) (simple security model)
 - Historic
- v3 (1998) Robust security
 - User/view based security (USM/VACM)
 - Full Internet Standard
- We will use SNMP v2c and v3 in this class

SNMP roles

- Terminology—We will be using Manager and Agent
- Manager (the monitoring station)
 - Sometimes known as the SNMP client
 - SNMPv3 calls it the Command Generator and Notification Receiver
- Agent (running on the equipment/server)
 - Sometimes known as the SNMP server
 - SNMPv3 calls it the Command Responder and Notification Originator

How does SNMP work?

Basic operators

- **get** (manager -> agent)
 - Query for a value
- **getnext** (manager -> agent)
 - Get next value (e.g. list of values for a table)
- **getresponse** (agent -> manager)
 - Response to **get**, **getnext**, or **set**, includes error returns
- **set** (manager -> agent)
 - Set a value, or perform an action
- **trap** (agent -> manager)
 - Spontaneous notification from equipment (line down, temperature above threshold, ...)

How does SNMP work?

- Query/response based
 - Monitoring generally uses **get, getnext, getbulk**
 - Changing state uses **set**
 - Response is always a **getresponse**
 - **getbulk** requires v2c or v3
- Notifications are delivered as **traps** or **informs**
 - **traps** are unacknowledged
 - **informs** are acknowledged (v2c, v3)
 - Use v2c format **traps**
 - No one uses **informs**

The SNMP database

- The information offered by a device is available in its Management Information Base (MIB)
 - SNMP uses Object Identifiers (OIDs) to organize this information
 - OIDs are keys to identifying each piece of data
 - OIDs are organized into a tree structure that is the MIB
 - MIB files document parts of the MIB on a device

OIDs

- **OID: Object Identifier**
 - A unique key to select a particular item of data in the device
 - The same piece of information is always found at the same OID. That's simple!
 - An OID is a variable-length string of numbers, e.g.
 - .1.3.6.1.2.1.1.3
- Allocated hierarchically in a tree to ensure uniqueness (*similar to DNS*)

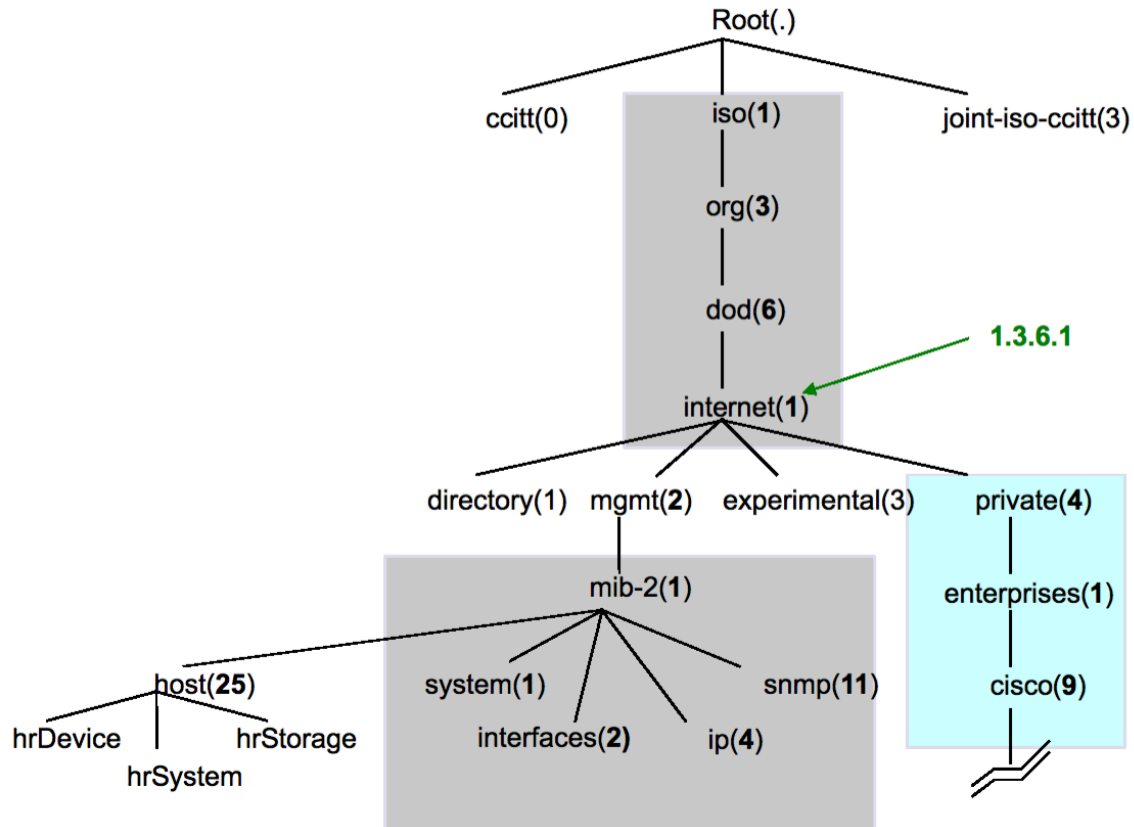
If Email Addresses were OIDs

- user@bdnog.org
 - *would have been something like:*
- user@bdnog.enterprises.private.internet.dod.org.iso
user@99999.1.4.1.6.3.1
 - except that we reverse the ordering, putting iso(1) first:
- .1.3.6.1.4.1.99999.117.115.101.114
 - Note the portion after 99999—it spells “user” in ascii dotted decimal!
- Don't worry about the deeply branched tree. What matters is that OIDs are unique.
 - Ensures vendors don't have conflicting OIDs
 - The numeric OID is what gets sent on the wire

OIDs and MIB files

- Read from left to right
OID components separated by '.'
 - .1.3.6.1.4.1.9. ...
- Each OID corresponds to a label
 - .1.3.6.1.2.1.1.5 => sysName
- The complete path:
 - .iso.org.dod.internet.mgmt.mib-2.system.sysName
- How do we convert from OIDs to Labels (and vice versa)?
- Use the MIBs files!

The MIB Tree



SNMP and Security

- SNMP versions 1 and 2c are insecure
- SNMP version 3 was created to fix this
- SNMPv3 authentication is based on a user
 - “User-based Security Model” (USM)
 - Authenticity and integrity
 - Keys are used for users and messages have digital signatures generated with a hash function (MD5 or SHA)
 - Privacy
 - Messages can be encrypted with secret-key (private) algorithms (DES or AES)
 - Temporary validity
 - Utilizes a synchronized clock with a 150 second window with sequence checking

SNMPv3 Security Levels

- **noAuthNoPriv**
 - No authentication, no privacy
- **authNoPriv**
 - Authentication with no privacy
- **authPriv**
 - Authentication with privacy

Cisco SNMP Configuration

- **Read-only**

- Cisco SNMP Configuration
- snmp-server community NetManage RO
- Enables SNMPv1 and v2c

```
snmp-server group ReadGroup v3 auth
snmp-server user admin ReadGroup v3 auth sha NetManage
```
- SNMPv3 authentication, no encryption

- **Read-write**

- ```
snmp-server group WriteGroup v3 auth write v1default
snmp-server user admin-rw WriteGroup v3 auth sha NetManage
priv aes 128 NetWrite
```
- Cisco allows authNoPriv and authPriv queries with this user
  - You could also define a read-write user without encryption (priv)
  - Note that we recommend using SNMP version 3 if you want write access using the **set** operator



# Net-SNMP Configuration

- Add a community string by editing `/etc/snmp/snmpd.conf` and adding:

```
rocommunity NetManage 10.10.0.0/16
```
- Add the SNMPv3 user

```
service snmpd stop
net-snmp-create-v3-user -a SHA -A NetManage
admin
service snmpd start
```
- Modify your user configuration file `~/.snmp/snmp.conf`, adding:

```
defVersion 3
defCommunity NetManage
defSecurityName admin
defSecurityLevel authNoPriv
defAuthPassphrase NetManage defAuthType SHA
```

# Querying an SNMP agent

- Using Net-SNMP command line tools...
- Some typical commands for querying:
  - `snmpget`
  - `snmpwalk`
  - `snmpbulkwalk` (requires v2c or v3)
  - `snmpstatus`
  - `snmptable`
- Syntax:
  - `snmpXXX -v1 -c<community> host [OID]`
  - `snmpXXX -v2c -c<community> host [OID]`
  - `snmpXXX -v3 -lauthNoPriv -u<user> -aSHA -A<pass> host [OID]`
- However, because you've setup the `snmp.conf` file, it's much easier  
`snmpxxx host [OID]`
- Or, if you want to force the version to v2c, for example:
  - `snmpxxx -v2c host [OID]`

# Querying an SNMP agent

- Let's look at some examples
  - `snmpstatus 10.10.0.254`
  - `snmpget 10.10.0.254 ifNumber.0`
  - `snmpwalk -v2c 10.10.0.254 ifDescr`

# Querying an SNMP agent

- Community:
  - A "security" string (password) to define whether the querying manager will have RO (read only) or RW (read write) access
  - This is the simplest form of authentication in SNMP
- OID
  - A value, for example, `.1.3.6.1.2.1.1.5.0`
  - or its name equivalent: `sysName.0`
- Let's ask for the system's name (using the OID above)
  - Why the `.0`? What do you notice?

# Queries Using snmp.conf

- Two walks:

```
snmpwalk 10.10.0.252 sysUpTime
```

```
DISMAN-EVENT-MIB::sysUpTimeInstance =
Timeticks: (1946738) 5:24:27.38
```

```
snmpwalk -v2c 3 10.10.0.252 sysUpTime
```

```
DISMAN-EVENT-MIB::sysUpTimeInstance =
Timeticks: (1953429) 5:25:34.29
```

- First walk used SNMPv3 as it was the default in snmp.conf, second walk specified SNMPv2c, and used the community string from snmp.conf.

# SNMP failure: no response?

- The device might be offline or unreachable  
The device might not be running an SNMP agent
- The device might be configured with a different community string
- The device might be configured to refuse SNMP queries from your IP address
- *In all of these cases you will get no response*

# SNMP Best Practices

- Secure your SNMP access and traffic:
  - Management VLAN
  - Access lists
  - Use SNMPv3 with authentication for queries and sets where possible
- Use SNMPv2c traps
  - Better formatted than v1 traps
  - Accurate timestamps
- Do no harm
  - Only poll as fast as you really need
  - Possible to drive CPU load on devices up and affect other protocol processing
  - It does no good to poll every 5 seconds if the device updates the counter every 10

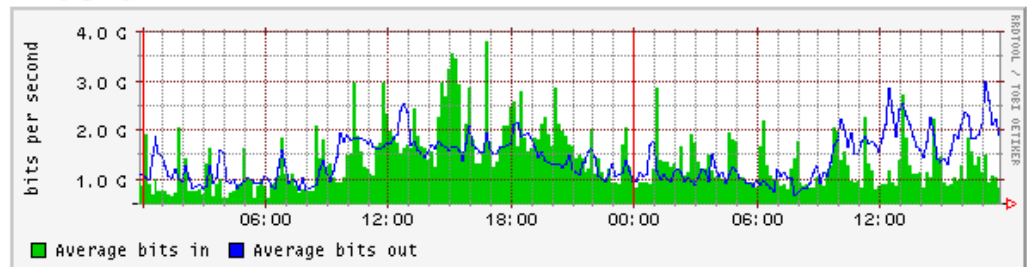
# Multi Router Traffic Grapher (MRTG)

The Multi Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network-links.

- MRTG generates HTML pages containing PNG images which provide an almost live visual representation of this traffic. Check <http://oss.oetiker.ch/mrtg/> for more information.
- From the mrtg pages:

*“You have a router, you want to know what it does all day long? Then MRTG is for you. It will monitor SNMP network devices and draw [pretty pictures](#) showing how much traffic has passed through each interface.”*

Daily graph





# MRTG continued

- MRTG has been the most common network traffic measurement tool for all Service Providers during this millenium.
- MRTG uses simple SNMP queries on a regular interval to generate graphs.
- External readers for MRTG graphs can create other interpretation of data.
- MRTG software can be used not only to measure network traffic on interfaces, but also build graphs of anything that has an equivalent SNMP MIB - like CPU load, disk availability, temperature, etc...
- Data sources can be anything that provides a counter or gauge value – not necessarily SNMP.
  - For example, graphing round trip times.

# MRTG issues

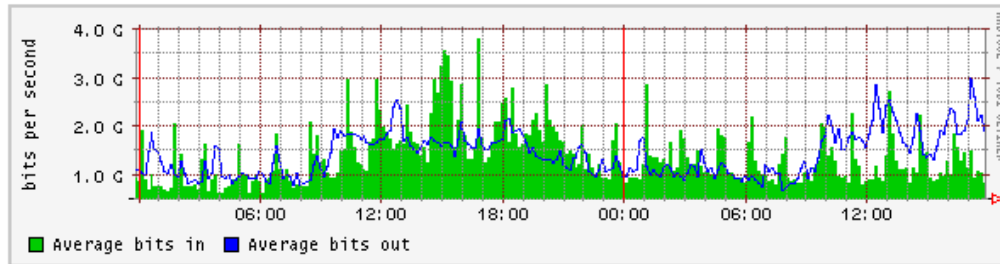
- MRTG generates each graph every 5 minutes. This can create considerable overhead if you are graphing for many devices (100's of routers with multiple interfaces for instance...).
  - Example: 500 routers, 2 interfaces each = 1000 graphs to generate. Potential CPU overhead.
- Very few customizable graphing options.
- MRTG management itself can be tedious work (see next slide...)

# Running MRTG

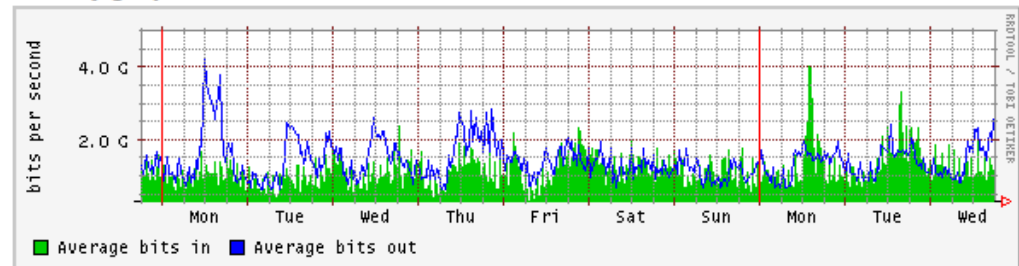
- Install or compile required packages
  - `apt-get install mrtg`
- Make `cfg` files for router interfaces with `cfgmaker`
- Create `html` pages from the `cfg` files with `indexmaker`
- Trigger MRTG periodically from `cron` or run it in daemon mode

# MRTG graphs

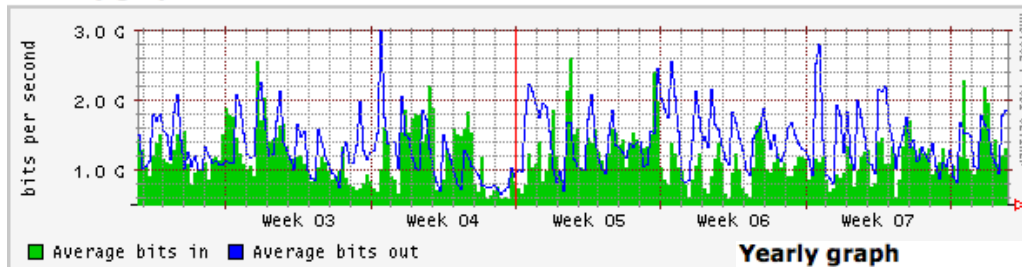
Daily graph



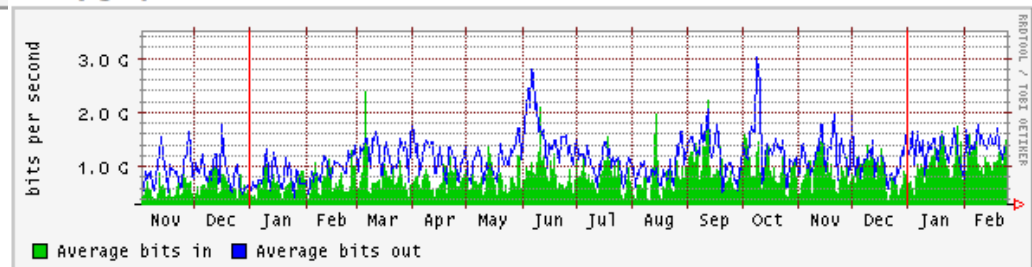
Weekly graph



Monthly graph



Yearly graph



Questions!