Network Switch Monitoring using SNMP

Kirti H. Wanjale[1], Jugal S. Patil[2], Sakshi Khode[3], Rutuja Chaudhary[4]
[1,2,3,4]*Vishwakarma Institute of Information Technology, Pune, Maharashtra*

Abstract **- In the present cloud emerging world, where every cloud service provider, providing IaaS, PaaS, or SaaS has a very important thing to look after and that is the monitoring of the infrastructure so that Reliability, Availability, and Serviceability of the product or service are taken care of. Private or public network switches is one the important component of such an infrastructure. Despite providing high speed and most secure service to the client if the network switches remain unmonitored then it could kill all the performance and security implied. In this paper, we are trying to demonstrate how Simple Network Management Protocol (SNMP) can be used to monitor the network switches.**

Index Terms **- Simple Network Management Protocol, switches, Linux virtual machines/servers, Network Manager, Agent, port, IP address, Terminal/Command Prompt, CPU usage, Memory Usage.**

I.INTRODUCTION

Throughout the IT industry Network Monitoring is a widespread term in today's world. All potential networking devices/components like routers, switches, servers, VM's, firewalls can be monitored for fault and monitoring. Continuous evaluation of these parameters is important to maintain the High Availability of the network. Monitoring again can be proactive and reactive, but efficient proactive monitoring of devices can prevent network failures or downtimes. Finding the fault like connection loss or disk volume full etc. will be considered as a reactive approach and actively looking for any performance bottlenecks could be a proactive approach. Here in this demonstration, we are trying to use both approaches to make a capable system.

Some common issues that might be encountered if switches or servers are not monitored are:

1.Poor Security

Unused or redundant backup ports are not monitored and kept disabled till their use then it could result in security breaches like virus or malware installation by an unknown or unauthorized individual.

2. Increased period of network downtime

As the health of the components is unknown the downtime or failures of the components cannot be mitigated.

3. Decreased lifespan of device

Hardware damage and corruption are possible due to frequent load, failures, or attacks. Thus, failing to monitor the devices will reduce their lifespan.

II.GOALS, FUNCTIONS OF NETWORK MONITORING

Network monitoring has a kind of server-client architecture, where Network Management Server (NMS) manages all aspects of the network.

- NMS can be connected to multiple devices/components for monitoring.
- NMS can pull the health status of the devices.
- NMS can be located in multiple segments of the network.
- Any kind of Fault or Overload can be detected by the NMS or sent to NMS by the devices.
- NMS can process the information from all the nodes to regulate the traffic and improve performance.
- NMS can set rules or change the configurations on the devices given such permission.

III.ISO NETWORK MANAGEMENT MODEL

International Organization of Standardization has given five major functional areas for network management in form of a model as follows,

Performance Management:

Network performance can be maintained by observing the available bandwidth on the node, load per job, throughput, CPU load, sensor values, etc. When analysis of such parameters is done at the NMS level, the cause of the performance issue can be detected and solved at the right time. The object of such monitoring

is to keep the heath and performance of the network in check.

Accounting Management:
Account Management is concerned with the users. Statistics of the user account can be maintained to observe their resource usage and information consumption. Based on this information, existing load on the resource per user, permissions to the user, existing cost per user/organization can be noted. Fairness of network access can also be maintained using accounting management.

Configuration Management:
Configuration Management implies control over the configurations related to both software and hardware on the server/device. For example, management of port and interface configurations on the device, DHCP, DNS controls, and other policies like firewall rules, etc. Description of interfaces can also be counted in configuration management.

Fault Management:
To keep the network available and running, detection, recognition, isolation are the important steps. Then these faults can be logged and corrected. Analysis of these faults can be used to predicts the faults that might occur in the future.

Security Management:
Network breaches, sabotages, whether intentional or otherwise can be avoided and kept track of by writing proper security policies for the network resources. Mapping of security rules to user sets, distribution across the devices, tracking security-related events are some of the included topics.

IV.SCENARIO

We are considering a scenario where an organization has its storage solution, and storage sites are distributed across the regions. It has some interface like s3 (simple storage service) through which one can access his data. The request for this data access will be redirected to the nearest proxy server to the client from there the organization needs to route this request to the correct site, node server, and drive enclosure where the data is stored. In this scenario irrespective of the data search, retrieval or write speed achieved in the storage solution, if the network performance is low due to network overload or connection issues and any such possible network problems, then the overall solution cannot succeed. In this scenario, the organization can have its own Network Management Servers to monitor all the crucial networks, devices, links, and performance so that the overall solution can achieve high availability, reliability, and accessibility by detecting such fault scenarios and taking appropriate action on them.

V.SIMPLE NETWORK MANAGEMENT PROTOCOL

Simple Network Management Protocol (SNMP) is an application–layer protocol defined by the Internet Architecture Board (IAB) for the management and monitoring of network-connected devices in Internet Protocol networks. SNMP collects data from different hardware and software, organizes the data, and helps in network fault detection and fault prevention analysis. Devices that support SNMP include cable modems, routers, switches, servers, workstations, printers, and more.
It allows devices to speak albeit the devices are different hardware and run different software. Without a protocol like SNMP, there would be no way for network management tools to spot devices, monitor network performance, keep track of changes to the network, or determine the status of network devices in real time. SNMP is nothing but a typical language that computers use to regulate one another and report important information.

SNMP Architecture:
Architecture of SNMP is based Client- Server model. Here the servers are SNMP Network Managers, and the clients are the agents so it is also called as Manager-Agent architecture.
SNMP Manager (NMS) – It is a client program which runs on remote machine (which when combined, is called the Network Management System) and queries agents through SNMP Get request and SNMP Set request in order to gather information like network status or to perform write operation. There can be multiple managers in a network.

SNMP Agents – It is a software program which runs on different SNMP enabled network devices such as

routers, switches, Wi-Fi, server, etc. It stores all data from its device into database (MIB) and collects it whenever manager requests.

MIB Database – A MIB (Management Information Base) is a database that specify the information that an agent shares with another agent. It follows SNMP Protocol. It has hierarchical structure.

SNMP Trap – As Managers can send commands to agents likewise agents can also send messages to Managers if it wants to Alert manager regarding any critical events without being polled.
In typical uses of SNMP, one or more administrative computers called managers have the task of monitoring or managing a group of hosts or devices on a computer network. Each managed system executes a software component called an agent which reports information via SNMP to the manager.

SNMP Versions:
Three versions of SNMP have been developed. SNMPv1 is the first version of the protocol. More recently SNMPv2c and SNMPv3 versions were developed, these versions provide advantages like improvements in performance, flexibility, and security.

MIB and OID:
MIBs and OIDs are the crucial parts of the SNMP architecture. These two components are vital for enabling you to watch network infrastructure and run troubleshooting.
The agent takes information from the MIB and passes it to the SNMP manager once a query has been made. This information contains status details about the connected device.
A MIB or Management Information Base may be a formatted document that resides within the SNMP manager designed to gather information and organize it into a hierarchical format. The SNMP manager uses information from the MIB to translate and interpret messages before forwarding to the end-user.
Resources stored during a MIB are mentioned as managed objects or management variables. MIB is just the central hub of knowledge inside the device. The MIB contains all of the performance data that is accessed at the loading of network monitoring tool.
There are many different managed objects In MIB which are identified by an Object Identifier or OID. An OID is an address that's want to differentiate between devices within the MIB hierarchy. The OID is preferably wont to navigate through variables and to ask unique characteristics on the SNMP device. The value of these identifiers can vary from text to numbers and counters.
These are often depicted as a tree. An OID is formatted during a string of numbers. (e.g., 1.3.6.1.4)

Protocols used in SNMP:
All the SNMP communication between devices and manager takes place using User Datagram Protocol (UDP), although SNMP falls under application layer of the Internet protocol suite.
SNMP manager can send the read/write request on port 161 of the agent. Manager can use any port for sending this request and it will receive the response from the agent on the same source port.
The agent forwards the traps or informs to the agent on port 162. Manager has responsibility to listen of this port. The agent may push this from any of its available port.
Ports 10161 and port 10162 are used instead of 161, 162 respectively if TLS (Trasport Layer Security) or DTL (Datagram Transport Layer Security) is used.
GetRequest, SetRequest, GetNextRequest, GetBulkRequest, Response, Trap, InformRequest and Report are the 8 PDU types supported by SNMP. GetBulkRequest and InformRequest were added in SNMPv2 and the Report PDU was added in SNMPv3.

VI.IMPLEMENTATION

Prerequisites:
We used a centos 7.7 virtual machine as a manager server or NMS.
Mellanox SN2100 switch and another centos 7.7 virtual machine to run the snmp agent.
So, the Functionality that is expected is the NMS server should be able to query any information to the agents, also allowed to set some values on the agents and agents should be able to send the trap/inform message to the NMS when certain triggers and triggered.
For this to work the machine that is chosen as an agent and as a manager must be supporting the SNMP

protocol. All our agents and NMS server support SNMP protocol.

The second requirement is that all the machines should have some SNMP software installed which will enable all the SNMP functionalities.

In our case, one of the agents was Mellanox switch which has its own flavor of Linux called Mellanox Onyx OS this operating system being Linux-based supports the Net-snmp Linux library that runs on the system as a daemon.

The other agent is centos 7.7 which is also a Linux distribution thus even this agent supports the Net-snmp library and can run snmp daemon.

When the NMS is concerned it is also a Linux machine so it also can utilize the Net-snmp library. The NMS has a bundle of functionality along with SNMP for the monitoring of other infrastructure. This software stack on the Manager machine is in python thus a python snmp module i.e. pysnmp and pyasn1 were chosen for this machine in production. For testing results using net-snmp are shown.

After installation of the software on all the machines now it is time to configure all the machines.

There are 3 prominent usable versions of SNMP v1, v2c, and v3

SNMP v1 is the most simple protocol which requires the agents to set different community strings and their respective read-write access. But this protocol has all communication in plain text with community string which is also transported in plain text.

SNMP v2c is the same as that of v1, it just adds the feature of informs along with trap. Informs when sent by the agent, are acknowledged by the manager by ack message.

SNMPv3 is the newest version of SNMP. Its management framework features primarily involve enhanced security.

The SNMPv3 architecture introduces the User-based Security Model (USM) for message security and the View-based Access Control Model (VACM) for access control.

SNMPv3 supports the SNMP "Engine ID" Identifier, which uniquely identifies each SNMP entity. Conflicts can occur if two entities have duplicate EngineID's. The EngineID is used to generate the key for authenticated messages.

Authentication and Encryption are 2 primary forms of SNMP v3 security models.

1. Authentication is used to ensure that traps are read by only the intended recipient. As messages are created, they are given a special key that is based on the EngineID of the entity. The key is shared with the intended recipient and used to receive the message.
2. Encryption encrypts the payload of the SNMP message to ensure that it cannot be read by unauthorized users. Any intercepted traps will be filled with garbled characters and will be unreadable. Privacy is especially useful in applications where SNMP messages must be routed over the Internet.

We started with the implementation of SNMP v1 and progressed to SNMPv3 since it is the most secure and reliable protocol to use.

Implementation details of the SNMP V3 protocol are presented below.

Configuration of Mellanox switch:

- for adding NMS (manager server) ipv4 addresses and creating a v3 user:
- and creating a v3 user:

```
>> net add snmp-server listening-address 10.230.242.110
>> net add snmp-server username cumulusro auth-sha seagate1 encrypt-aes encryptseagate1.
## username and passphrases can be changed
>> net add snmp-server username cumulusro auth-sha seagate1 encrypt-aes encryptseagate1 oid .1
## username and passphrases should be the same as that configured in the previous command
>> net commit
```

- for enabling traps/informs:

```
>> net add snmp-server trap-destination 10.230.242.110 username infom_sender auth-sha authpass encrypt-aes privpass engine-id 0x80001f8880ec70e17424be1f5f00000000 inform
## username, passphrase, and engine-id should be as mentioned as it is configured on the server and it should
## match with user config on the server
>> net add snmp-server trap-link-up check-frequency 15
>> net add snmp-server trap-link-down check-frequency 10
>> net add snmp-server trap-snmp-auth-failures
>> net add snmp-server trap-cpu-load-average one-minute 4.34 five-minute 2.32 fifteen-minute 6.5
>> net commit
```

Configuration of management server :

- installation of packages :

```
>> sudo yum install net-snmp net-snmp-libs net-snmp-utils -y
```

- configuration for traps :

```
>> systemctl stop snmptrapd
## add following lines in /etc/snmp/snmptrad.conf
>> createUser inform_sender SHA authpass AES privpass
>> authUser log,execute,log inform_sender
## save the file and exit
>> systemctl start snmptrapd
```

- SNMP queries :

```
>> systemctl start snmpd
## Check if firewalld is running or not by the following command
>> systemctl status firewalld
## if it is running then run 2 commands mentioned below else skip these commands
>> firewall-cmd --permanent --zone=public --add-port={161/udp,162/udp}
>> firewall-cmd --reload
##now we are ready to do snmp v3 queries
>> snmpget -v3 -u cumulusro -a SHA -A seagate1 -x AES -X encryptseagate1 -l authPriv 10.237.66.62 sysDescr.0
SNMPv2-MIB::sysDescr.0 = STRING: Cumulus Linux 3.7.10 (Linux Kernel 4.1.33-1+cl3u24)
>> snmpwalk -v3 -u cumulusro -a SHA -A seagate1 -x AES -X encryptseagate1 -l authPriv 10.237.66.62 interfaces
IF-MIB::ifNumber.0 = INTEGER: 37
IF-MIB::ifIndex.1 = INTEGER: 1
IF-MIB::ifIndex.2 = INTEGER: 2
IF-MIB::ifIndex.3 = INTEGER: 3
IF-MIB::ifIndex.4 = INTEGER: 4
IF-MIB::ifIndex.17 = INTEGER: 17
IF-MIB::ifIndex.18 = INTEGER: 18
IF-MIB::ifIndex.19 = INTEGER: 19
IF-MIB::ifIndex.20 = INTEGER: 20
IF-MIB::ifIndex.21 = INTEGER: 21
IF-MIB::ifIndex.22 = INTEGER: 22
IF-MIB::ifIndex.23 = INTEGER: 23
IF-MIB::ifIndex.24 = INTEGER: 24
IF-MIB::ifIndex.25 = INTEGER: 25
. . . . . .
output is cut since it's too long.
```

Different traps triggers and their configurations on Cumulus Linux
Traps configured directly through NCLU commands

1. Link Up trap:

```
>> net add snmp-server trap-link-up check-frequency 15
```

2.Link Down trap:

```
>> net add snmp-server trap-link-down check-frequency 10
```

3.SNMP Authentication Failure :

```
>> net add snmp-server trap-snmp-auth-failures
```

4.CPU load average exceeding the threshold :

```
>> net add snmp-server trap-cpu-load-average one-minute 4.34 five-minute 2.32 fifteen minute 6.5
```

Other types of traps are supported by net-snmp and can be enabled by editing the snmpd.conf file
##All the following commands are to be placed inside the snmpd.conf file
SNMPv3 needs an internal v3 user with appropriate permissions to internally query necessary information

```
>> createuser internaluser
>> iquerysecname internaluser
>> rouser internaluser
```

Commands that set different trap triggers :

5.Generate alert if any cpu core usage exceeds 90%

```
>> monitor -r 60 "VM CPU Load Too High" hrProcessorLoad > 90
```

6.Generate an alert when the 1 minute, 5 minute, and 15-minute load averages exceed a certain amount

```
>> load 12 10 5
>> monitor -r 60 -o laNames -o laErrMessage "Load Average Exceeded" laErrorFlag != 0
```

7.Generate an alert when the free space on the folders to be monitored falls below the minimum space required.

```
>> disk /var/log 10%
#OR
>> includeAllDisks 5%
>> monitor -r 60 -o dskErrorMsg "Folder HDD Space Low" dskErrorFlag != 0
```

8.Generate an alert when VM RAM used exceeds X KB. e.g. 4 GB = 4194304 KB

```
>> monitor -r 60 -I "VM RAM Usage Too High" hrStorageUsed.1 > 4194304
```

9. Generate an alert when any process memory usage exceeds X KB, e.g. 1 GB = 1048576 KB

```
>> monitor -r 60 -o hrSWRunName -o hrSWRunPath "Process RAM Usage Too High" hrSWRunPerfMem > 1048576
```

10. Generate traps for ENTITY-MIB and ENTITY-SENSOR-MIB

```
## We can monitor the operational status of ENTITY-MIB and ENTITY-SENSOR-MIB using oid's as shown below
>> monitor -I -r 10 -o 1.3.6.1.2.1.47.1.1.1.1.7.100011001 "Fan1 Not OK" 1.3.6.1.2.1.99.1.1.1.5.100011001 > 1
```

```
## But we can use the oid name if the 'snmp-mibs-downloader' is installed.
# Open /etc/apt/sources.list in a text editor.
# Add the non-free repository, then save the file:
  >> sudo deb [http://ftp.us.debian.org/debian/] buster main non-free
# Update the switch:
  >> sudo -E apt-get update
# Install the snmp-mibs-downloader:
  >> sudo -E apt-get install snmp-mibs-downloader
# Open the /etc/snmp/snmp.conf file to verify that the mibs : line is commented out:
  >> #mibs :
# Open the /etc/default/snmpd file to verify that the export MIBS= line is commented out:
  >> #export MIBS=
# After you confirm the configuration, remove or comment out the non-free repository in /etc/apt/sources.list.
>> #deb [http://ftp.us.debian.org/debian/] buster main non-free
## Now we can write the command with oid name as
>> monitor -I -r 10 -o entPhysicalName.100011001 "Fan1 Not OK" entPhySensorOperStatus.100011001 > 1
```

11. Generate traps for temperature sensors

```
>> monitor -r 500 lmTemSensor -o lmTempSensorsDevice lmTempSensorsValue > 68000
```

12. Free Memory Notifications

```
>> monitor MemFreeTotal -o memTotalReal memTotalFree < 1000000
```

13. Monitoring of Fan speed

```
>> monitor -r 100 lmFanSensor -o lmFanSensorsDevice lmFanSensorsValue < 7000
```

Implementation of Automatically Triggered Informs on Cumulus Linux Switch

- ColdStart inform on restart of snmpd.service

  Command on a switch :

  ```
  >> systemctl restart snmpd
  ```

  Output log on the server

  ```
  Aug 7 02:22:40 ssc-vm-0804 snmptrapd[23361]: 2020-08-07 02:22:40 mlx1-r18.pun.seagate.com [UDP: [10.237.66.62]:54490->[10.230.242.110]:162]:#012DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (18) 0:00:00.18#011SNMPv2-MIB::snmpTrapOID.0 = OID: SNMPv2-MIB::coldStart#011SNMPv2-MIB::snmpTrapEnterprise.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
  ```

- Temperature exceeded beyond threshold inform

  Config on switch

  ```
  monitor -r 60 -o lmTempSensorsDevice "Temperature above threshold" lmTempSensorsValue > 25000
  ```

  Output log on the server

  ```
  Aug 7 03:09:58 ssc-vm-0804 snmptrapd[23361]: 2020-08-07 03:09:58 mlx1-r18.pun.seagate.com [UDP: [10.237.66.62]:47774->[10.230.242.110]:162]:#012DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (27) 0:00:00.27#011SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired#011DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Temperature above threshold#011DISMAN-EVENT-MIB::mteHotTargetName.0 = STRING: #011DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: #011DISMAN-EVENT-MIB::mteHotOID.0 = OID: LM-SENSORS-MIB::lmTempSensorsValue.22#011DISMAN-EVENT-MIB::mteHotValue.0 = Wrong Type (should be INTEGER): Gauge32: 32000#011LM-SENSORS-MIB::lmTempSensorsDevice.22 = STRING: spectrum-i2c-1-48:temp1
  Wrong type (should be INTEGER) error should be due to Guage32 type of Sensor Value
  ```

- Fan Speed crossed the limit inform

  Config on switch

  ```
  monitor -r 60 -o lmFanSensorsDevice "Fan speed exceeded above threshold" lmFanSensorsValue > 17000
  ```

  Output on server

  ```
  Aug 7 03:09:58 ssc-vm-0804 snmptrapd[23361]: 2020-08-07 03:09:58 mlx1-r18.pun.seagate.com [UDP: [10.237.66.62]:47774->[10.230.242.110]:162]:#012DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (27) 0:00:00.27#011SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired#011DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: Fan speed exceeded above thres#011DISMAN-EVENT-MIB::mteHotTargetName.0 = STRING: #011DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: #011DISMAN-EVENT-MIB::mteHotOID.0 = OID: LM-SENSORS-MIB::lmFanSensorsValue.21#011DISMAN-EVENT-MIB::mteHotValue.0 = Wrong Type (should be INTEGER): Gauge32: 17396#011LM-SENSORS-MIB::lmFanSensorsDevice.21 = STRING: fan4
  ```

- Inform for CPU average load exceeded the threshold

  Config on switch

  ```
  >> load 5.0 2.0 1.0
  >> monitor -r 60 -o laNames -o laErrMessage "laTable" laErrorFlag != 0
  ```

Output on server log

```
Aug 7 03:25:22 ssc-vm-0804 snmptrapd[23361]: 2020-08-07 03:25:22 mlx1-r18.pun.seagate.com [UDP: [10.237.66.62]:52803->[10.230.242.110]:162]:#012DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (21) 0:00:00.21#011SNMPv2-MIB::snmpTrapOID.0 = OID: DISMAN-EVENT-MIB::mteTriggerFired#011DISMAN-EVENT-MIB::mteHotTrigger.0 = STRING: laTable#011DISMAN-EVENT-MIB::mteHotTargetName.0 = STRING: #011DISMAN-EVENT-MIB::mteHotContextName.0 = STRING: #011DISMAN-EVENT-MIB::mteHotOID.0 = OID: UCD-SNMP-MIB::laErrorFlag.3#011DISMAN-EVENT-MIB::mteHotValue.0 = INTEGER: 1#011UCD-SNMP-MIB::laNames.3 = STRING: Load-15#011UCD-SNMP-MIB::laErrMessage.3 = STRING: 15 min Load Average too high (= 1.00)
```

VII.CONCLUSION

In this demonstration, we tried to show how powerful SNMP protocol is, and how easy it is to set up a basic management/monitoring infrastructure for securing network performance. Along with power comes the great responsibility although SNMPv3 provides sufficient security, the admin should be aware of all read/write permissions given to an NMS and security of vulnerabilities of the NMS. Further, along with fault detection through traps and information polling, we can make use of the same information to train a basic AI/ML model to predict future failures, prevent performance bottlenecks and node downtimes.

ACKNOWLEDGMENT

We sincerely acknowledge our indebtedness to our mentor Prof. Vidya Gaikwad Madam for the immense support and the mentorship that she provided. I express my sincere gratitude to Prof. Kirti Wanjale Mam for her guidance and the help regarding paper writing and publishing. We can never forget our parents and family members whose blessings and best wishes inspired us in the journey.

REFERENCES

[1] The OSI Network Management Model- Capacity and performance management. Nuangjamnong C., Maj S.P., Veal D. (2008) Proceedings of the 4th IEEE International Conference on Management of Innovation and Technology, ICMIT, art. no. 4654552, pp. 1266-1270.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Shaffi, Abubucker & Al-Obaidy, Mohaned. (2013). MANAGING NETWORK COMPONENTS USING SNMP. International Journal of Scientific Knowledge.

[3] Behrouz A. Forouzan “Data Communication and Networking”, 5th Ed, Tata McGraw Hills.

[4] John Cowley, “Communications and Networking: An Introduction”, 2007, Springer.

[5] L. Yang and G. Michailidis, “Sampled based estimation of network traffic flow characteristics,” in INFOCOM 2007, 26th IEEE International Conference on Computer Communications, pp. 1775–1783, May 2007

[6] Larry L. Peterson S. Davie, “Computer Networks A System Approach”, 4th Ed, Morgan Kaufmann.

[7] Hossein Bidgoli, “The Handbook of Computer Networks”, Volume-2, 5th Ed, Wiley.

[8] Jonathan Saperia, “SNMP at the Edge: Building Effective Service Management Systems

[9] Nuangjamnong, C., Maj, S. P., & Veal, D. R. (2008). The OSI Network Management Model - Capacity and performance management. Proceedings of 4th IEEE International Conference on Management of Innovation and Technology. ICMIT 2008. (pp. 1266-1270). Bangkok, Thailand. IEEE.

[10] Net-SNMP: Wikipedia [Online] https://en. wikipedia.org/wiki/Net-SNMP

[11] Project: CORTX-Monitor (SNMP sensor implementation) [Online] https://raw.githubuser content.com/Seagate/cortx-monitor/main/low-level/sensors/impl/generic/SNMP_traps.py

[12] The Net-SNMP Wiki [Online] - http://www.net-snmp.org/wiki/index.php/Main_Page