INVESTIGATING THE PERFORMANCE OF A SECURE RASPBERRY PI SNMPV3 AGENT

Cristian STANICA, Valeriu Manuel IONESCU
University of Pitesti
Department of Electronics, Communications and Computers
Pitesti, Romania
valeriu.ionescu@upit.ro

Keywords: Raspberri Pi, SNMP v3, performance.

Abstract: *The Raspberry Pi is a single-board computer that can be used in many network related operations due to its low cost, size and high versatility. SNMP protocol is used to monitor dedicated network devices and systems both Windows and Linux. Raspberry Pi can be used to monitor the network as a SNMP agent. There are, however, two aspects that may have a performance impact on a low computing power system: encryption and the polling interval. SNMP version 3 is the recommended SNMP version and allows the use of encrypted messages to increase the network management security but needs more processing power. It is also possible to shorten the polling interval for the SNMP information in order to better detect fast changing network characteristics, but this can become a stressful operation. This paper will test the usability of Raspberry Pi as a SNMPv3 agent with encryption support for default and fast refresh interval.*

1. INTRODUCTION

Simple Network Management Protocol (SNMP) is a standard internet protocol used for administrating equipments in IP networks. The equipments which support SNMP are routers, switches, servers, printers, modems, racks, and other. SNMP is a component of the Internet Protocol Suite which was defined by Internet Engineering Task Force (IETF). It consists of a set of standards for managing the network, including the application layer protocol, a diagram of the database and a set of data objects. SNMP [1, 2] exposes the management data in the form of variables in administration systems which describe the system's configuration. These variables can then be interrogated by the management applications.

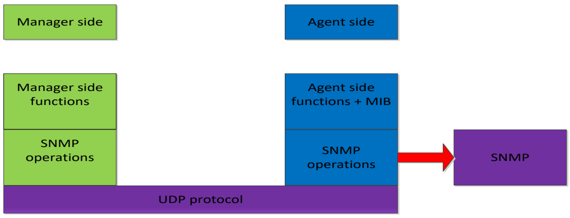


Fig.1 The structure of SNMP management system

1.1 Evolution of SNMP

SNMPv1 uses a community string to establish connection between an agent and one or more managers, this prevents unwanted administrators to read or write data on the agent's side. The community name is placed in every SNMP request, it is not encrypted or encoded. There are two ways of access: *read-only and read-write.*

SNMPv2 revises version 1 and includes enhancements in performance, security, and confidentiality and manager-to-manager communications. It introduced *GetBulkRequest*, to capture larger quantities of management data in a single request. This version was considered too complex and two other versions derived from it: SNMPv2c, which doesn't include the complex security models, using instead the security community string of SNMPv1 and SNMPv2u, which tries to offer a higher security than SNMPv1 but without the great complexity of SNMPv2.

SNMPv3 adds security features and remote SNMP management accessories. Each message contains security parameters that are encoded as a string of bytes. SNMPv3 offers important security features like the encryption of packets to prevent snooping from an unauthorized source, integrity and authentication.

The protocols used for Authentication are MD5 and SHA and for Privacy DES (Data Encryption Standard) and AES (Advanced Encryption Standard). SNMP Research products beginning with release 16.1 (domestic) also support the 3DES (Triple DES) privacy protocol. In 2004 IETF recognized SNMPv3 defined by RFC 3411-RFC3418 as the current standard version of SNMP. Previous versions are considered obsolete [3].

1.2 SNMP vs. WMI

Windows Management Instrumentation (WMI) is a management technology used on all recent versions of Windows. WMI offers a powerful set of services, including event notification and query-based information retrieval. It can be used to manage local or remote computers.

SNMP compared to WMI is extremely efficient in its use of CPU, RAM and bandwidth (on both manager and agent, WMI uses approximately 5 times the resources required to poll the same object on the same frequency), it can be implemented on all computer platforms (unlike WMI for Windows systems only), and its implementation is simpler, it is up to us to decide how much or how little information the SNMP protocol will expose. However, WMI, although is much more complex its information is richer, because of this, an attacker might find it more useful to intercept the packets, WMI's main advantage is that it uses real reboot time for uptime metrics giving a much more realistic and nicer monitoring data, due to this fact WMI is considered more powerful.

2. OPTIMIZING SNMP

In its basic operation, SNMP users, called managers, have the task to monitor a group of hosts, or equipments in a private network. Each managed system executes, at the same time, a software component called agent which sends information from the host to the manager. The agent reveals its data under different variable types. These variables are organized hierarchical and are described in the Management Information Bases (MIBs).

A SNMP monitored network consists of three components:

- the monitored equipments;
- the agent – the software which runs on the monitored equipments;
- network management system (NMS) – the software which runs on the administrator side.

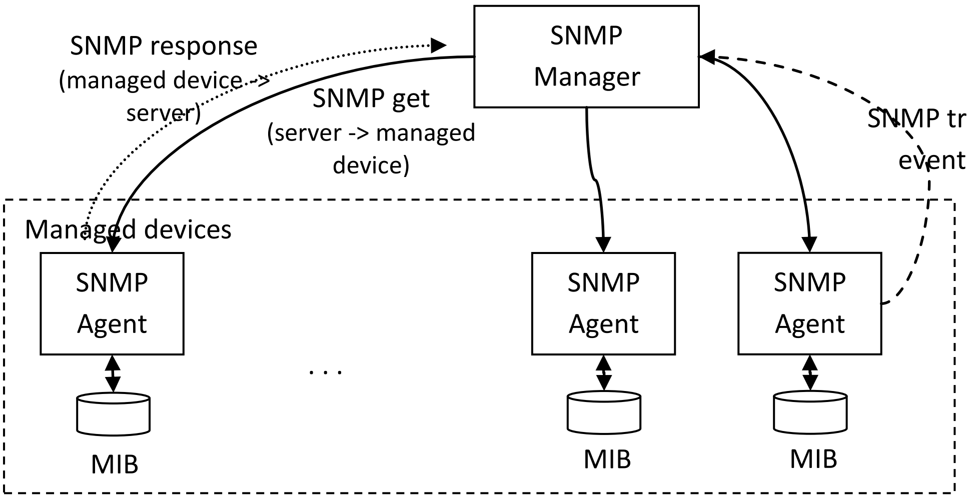


Fig.2 A monitored network using SNMP. Usually the server queries the devices, but devices can also generate messages to the server when a SNMP trap is configured

The default polling interval for SNMP is 5 minutes, this means that every 5 minutes the manager sends SNMP commands to query the agents about different information in their MIBs, the agent responds to the manager with the required information, this relatively high interval causes the polling to be somewhat inaccurate because if we want to gather relatively sensitive data it might be more useful to do a 1 sec polling or even faster. Faster polling means that we can view greater, smoother results and we can understand the flow of different monitored information of the system better. This may lead however to higher CPU and RAM usage.

We will be using two machines, one will contain the manager side (or server side) and one will contain the agent side (or client side), both of them will be using Ubuntu 14.04.

On the server side:

We will first update the apt database using the command:

sudo apt-get update

Then we will install the SNMP manager:

sudo apt-get install snmp

On the client side:

First update the application database then install the SNMP agent using the commands:

sudo apt-get update

sudo apt-get install snmpd

And that's all there is regarding installation, now we can move to the next step and configure the agent side.

2.1 Configuring the client

The server side doesn't require additional configuration for its basic functionality, on the other hand on the client side our configuration will be fairly easy, and we only need to configure a file [4, 5]:

sudo gedit /etc/snmp/snmpd.conf – this command directly opens the file snmpd.conf with ubuntu's text editor, and we can configure it because we ran sudo privileges.

In this file we will have to find the agentAddress lines and change it from:

```
   #  Listen for connections from the local
system only
       agentAddress  udp:127.0.0.1:161
   #  Listen for connections on all
interfaces (both IPv4 *and* IPv6)
   #  agentAddress udp:161,udp6:[::1]:161
```

To:

```
   #  Listen for connections from the local
system only
   #  agentAddress  udp:127.0.0.1:161
   #  Listen for connections on all
interfaces (both IPv4 *and* IPv6)
       agentAddress udp:161,udp6:[::1]:161
```

Commenting the second line and uncommenting the fourth line will enable the agent to listen on all interfaces (not just on its local system) for SNMP queries and respond to them.

Next, we will create a user by inserting the following line:

createUser demouser MD5 demopassword DES demoencryption

This command creates the user *demouser* with MD5 authentication type (we can also use SHA), with the passphrase *demopassword* (must be at least 8 characters in length), with DES privacy protocol (we can also use AES), and with the privacy protocol passphrase *demoencryption.*

Below this line we will add another line to specify the level of access that this user will have:

rwuser demouser priv

This command gives the user demouser read-write access (we can also assign him only read-only access with the command *rouser demouser priv*), and enforces encryption by stating priv at the end of the line. If we wanted to restrict a specific user to a specific level of OID we can specify the highest level OID that the user should have access to after the priv statement at the end of the line.

After implementing these lines we have to save our configuration file by typing *:w*, and then we will exit by typing *:x*.

For the SNMP service to begin working we will have to restart it in order for the changes to take effect using the following command:

sudo service snmpd restart

We can then test our client-server configuration by going at the server side and typing a snmp command, for example the snmpget command:

snmpget -u demouser -l authPriv -a MD5 -x DES -A demopassword -X demoencryption remote_host 1.3.6.1.2.1.1.1.0

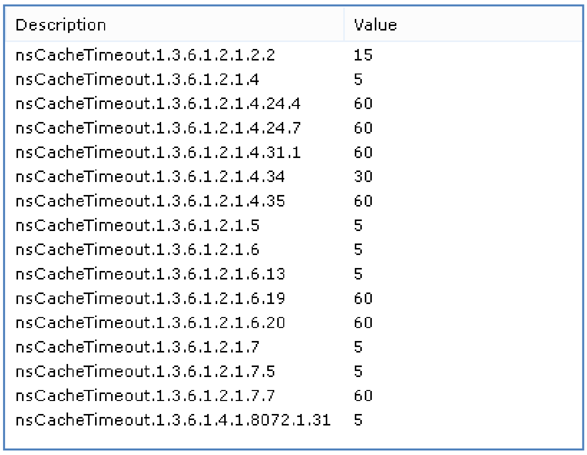
Which will display the following information:

SNMPv2-MIB::sysDescr.0 = STRING: Linux target 3.13.0-24-generic #46-Ubuntu SMP Wed Dec 23 22:11:08 UTC 2015 x86_64

We can now conclude that we have a running client-server configuration.

2.2 Setting the agent update interval

The default update interval for the SNMP agent is 15 seconds on the "ifTable" branch and 5 seconds on the "ip" branch as we can see in the figure below (first row and second row). These branches are a part of the "nsCacheTable" (1.3.6.1.4.1.8072.1.5.3) which contains entries that refer to, as the MIB describes, "The length of time (in seconds) for which the data in this particular entry will remain valid".



Description	Value
nsCacheTimeout.1.3.6.1.2.1.2.2	15
nsCacheTimeout.1.3.6.1.2.1.4	5
nsCacheTimeout.1.3.6.1.2.1.4.24.4	60
nsCacheTimeout.1.3.6.1.2.1.4.24.7	60
nsCacheTimeout.1.3.6.1.2.1.4.31.1	60
nsCacheTimeout.1.3.6.1.2.1.4.34	30
nsCacheTimeout.1.3.6.1.2.1.4.35	60
nsCacheTimeout.1.3.6.1.2.1.5	5
nsCacheTimeout.1.3.6.1.2.1.6	5
nsCacheTimeout.1.3.6.1.2.1.6.13	5
nsCacheTimeout.1.3.6.1.2.1.6.19	60
nsCacheTimeout.1.3.6.1.2.1.6.20	60
nsCacheTimeout.1.3.6.1.2.1.7	5
nsCacheTimeout.1.3.6.1.2.1.7.5	5
nsCacheTimeout.1.3.6.1.2.1.7.7	60
nsCacheTimeout.1.3.6.1.4.1.8072.1.31	5

Fig.3 The default net-snmp installation table

If these interval suit your needs, you are good to go but if you want to see more detailed data from the polling we can set these intervals lower for faster polling rates to better smoothen the graphical data, however this faster polling rate will increase our CPU and ram usage on the agent but will give us much more information about our managed equipment enabling this way for SNMP to be a better monitoring tool, comparable to WMI.

We can use the snmpset command to modify these parameters as below:

snmpset –v 3 192.168.2.1 1.3.6.1.4.1.8072.1.5.3.1.2.1.3.6.1.2.1.2.2 i 1

where:

-snmpset : SNMP command used on the server machine;

-v 3 : snmp version;

-192.168.2.1 : ip of target machine with the agent installed on it;

-1.3.6.1.4.1.8072.1.5.3.1.2.1.3.6.1.2.1.2.2 : OID of the MIB to modify the update interval;

- i 1 : set integer value of 1 (updating every second).

3. MONITORING SNMPv3

Raspberry Pi is a mini computer, ranging from 5 $ - 35 $ having the approximate size of a credit card, that can be plugged into a pc monitor or a TV and will behave just like a usual computer if we add a keyboard and a mouse to it. It is designed for all people of all ages to help them explore computing at a low cost, and can be used for anything, from browsing the internet to interacting with the real world.

Fig.4 The default starter kit for Raspberry Pi Zero

Although the Raspberry Pi Zero is not as powerful as the Raspberry Pi 2 Model B, it may suite our needs, mainly we can use it as a monitoring system in a lan network.

Because it runs Raspbian we can easily install SNMP on it to poll our agents that are installed on the network, this will help network engineers to lower the cost of monitoring equipments, due to its low cost, and it has good performances to replace the current monitoring systems.

3.1. Management Information Base (MIB)

A management information base is a database used for better handling entities in a communications network. It is often associated with SNMP, and it refers to a complete collection of monitored information available on an entity, it is often used to describe a certain subset.

Objects in the MIB database are defined using Abstract Syntax Notation 1 (ASN.1), and are identified by the O.I.D. (Object

Identifier). The database is hierarchical and the objects are called using certain snmp commands.

Example O.I.D.:

1.3.6.1.4.1.2682.1.4.5.1.1.99.1.6

Example SNMPv3 commands: GetRequest, SetRequest, GetNextRequest, GetBulkRequest, Response, Trap, InformRequest.

3.3. Monitoring using MRTG

MRTG (Multi Router Traffic Grapher) is a tool which we can use to monitor our load on the Raspberry Pi 2, mainly CPU and RAM usage, to see how well this board can sustain SNMPv3. MRTG generates HTML pages which provide a real time visualization of this load [6].

On a newly installed version of Ubuntu we will need to install the Web Server (apache2):

sudo apt-get install apache2

Then we can install MRTG:

sudo apt-get install mrtg

We will also update the apt database using the command:

sudo apt-get update

We will then have to configure our /etc/mrtg/mrtg.cfg file:

sudo gedit /etc/mrtg/mrtg.cfg

Using this minimal script and adding the IP of the Pi and the O.I.D's for the CPU and RAM usage plus the SNMPv3 specific options:

```
   WorkDir: /var/www/mrtg
   Target[raspberry]: O.I.D.:public@x.x.x.x
   MaxBytes[raspberry]: 100
   Title[raspberry]:    Load    Analysis    on
Raspberry Pi 2
   PageTop[raspberry]:                <H1>CPU
Utilization</H1>
```

We can successfully analyze SNMPv# on our Pi, at least in theory.

After carefully reading a lot of documentation on MRTG and its use with SNMPv3, we are able to successfully plot the CPU and RAM usage of the board.

After configuring the cfg file we can start our MRTG process with the following command:

sudo env LANG=C /usr/bin/mrtg /etc/mrtg/mrtg.cfg

We can then plot our results using the syntax:

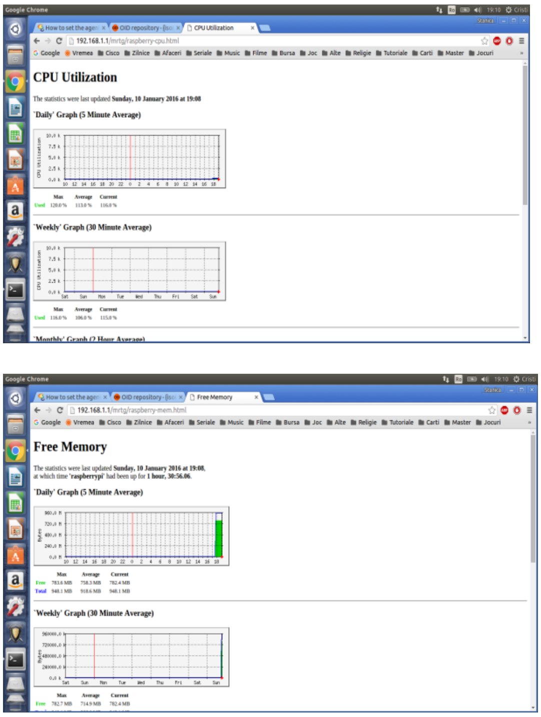
sudo indexmaker –output=/var/www/mrtg/index.html /etc/mrtg/mrtg.cfg

This will generate our nice HTMP graphs from which we can acquire information about the board and its behavior in a SNMPv3 monitored network.

4. TESTING THE SNMP MONITORING SETUP

Our standard test includes generating a constant ICMP traffic of 64000 bytes from a PC to the Pi for an hour each test and monitoring the Pi under different refresh rates to see if it can handle a faster polling rate of the agent and how many devices can it monitor in a network.

First we will monitor the default settings of the nsCacheTable.



CPU Utilization

The statistics were last updated **Sunday, 10 January 2016 at 19:08**

'Daily' Graph (5 Minute Average)

	Max	Average	Current
Used	120.0 %	113.0 %	116.0 %

'Weekly' Graph (30 Minute Average)

	Max	Average	Current
Used	116.0 %	106.0 %	115.0 %

'Monthly' Graph (2 Hour Average)

Free Memory

The statistics were last updated **Sunday, 10 January 2016 at 19:08**, at which time **'raspberrypi'** had been up for **1 hour, 30:56.06**.

'Daily' Graph (5 Minute Average)

	Max	Average	Current
Free	783.6 MB	758.3 MB	782.4 MB
Total	948.1 MB	918.6 MB	948.1 MB

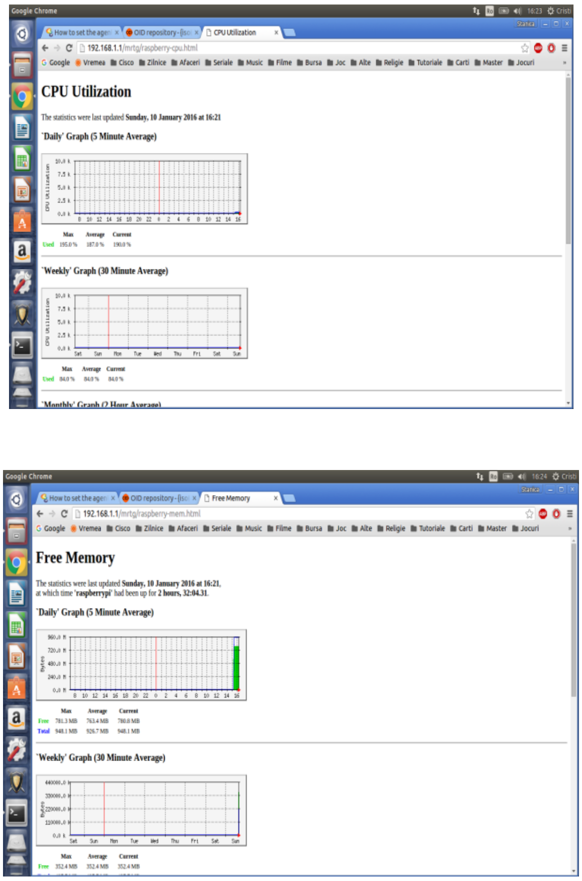
'Weekly' Graph (30 Minute Average)

	Max	Average	Current
Free	782.7 MB	714.9 MB	782.4 MB

Fig.5 The default test values of the nsCacheTable

Then we will change the agent refresh rate as explained before. We will change all the values to 1 so that the information the agent polls from the Pi will be every second. Results are in Fig.6.

And finally we will set all the values to 0. Results are in Fig.7.



CPU Utilization

The statistics were last updated **Sunday, 10 January 2016 at 16:21**

'Daily' Graph (5 Minute Average)

	Max	Average	Current
Used	195.0 %	187.0 %	190.0 %

'Weekly' Graph (30 Minute Average)

	Max	Average	Current
Used	84.0 %	84.0 %	84.0 %

'Monthly' Graph (2 Hour Average)

Free Memory

The statistics were last updated **Sunday, 10 January 2016 at 16:21**, at which time '**raspberrypi**' had been up for **2 hours, 32:04.31**.

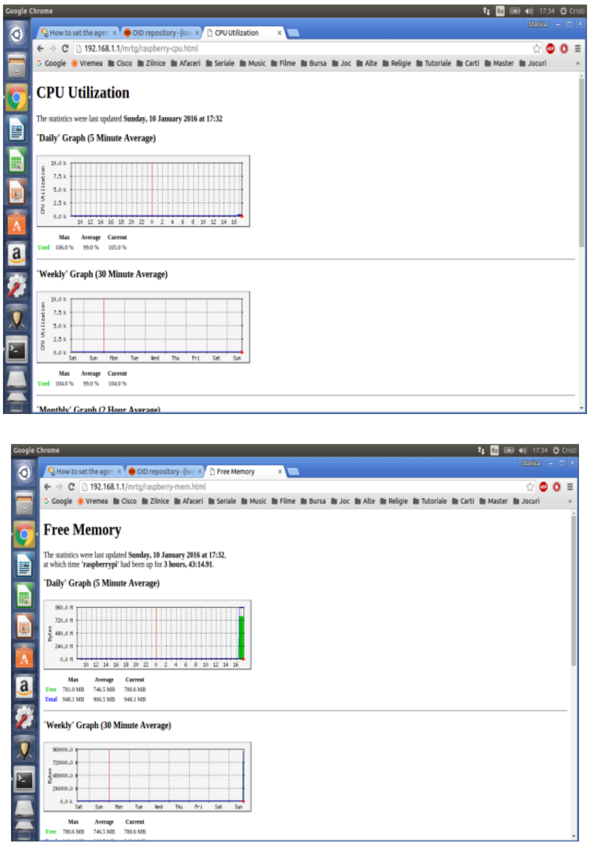
'Daily' Graph (5 Minute Average)

	Max	Average	Current
Free	781.3 MB	763.4 MB	780.8 MB
Total	948.1 MB	926.7 MB	948.1 MB

'Weekly' Graph (30 Minute Average)

	Max	Average	Current
Free	352.4 MB	352.4 MB	352.4 MB

Fig.6 The modified test values of the nsCacheTable for value 1



CPU Utilization

The statistics were last updated **Sunday, 10 January 2016 at 17:32**

'Daily' Graph (5 Minute Average)

	Max	Average	Current
Used	106.0 %	99.0 %	105.0 %

'Weekly' Graph (30 Minute Average)

	Max	Average	Current
Used	104.0 %	99.0 %	104.0 %

'Monthly' Graph (2 Hour Average)

Free Memory

The statistics were last updated **Sunday, 10 January 2016 at 17:32**, at which time '**raspberrypi**' had been up for **3 hours, 43:14.91**.

'Daily' Graph (5 Minute Average)

	Max	Average	Current
Free	781.0 MB	746.5 MB	780.6 MB
Total	948.1 MB	906.5 MB	948.1 MB

'Weekly' Graph (30 Minute Average)

	Max	Average	Current
Free	780.6 MB	746.5 MB	780.6 MB

Fig.7 The modified test values of the nsCacheTable for value 0

Table 1 Monitoring CPU and memory usage for the SNMPv3 agent

Modified to 1			
CPU	**max**	**average**	**current**
	1,95 %	**1,87 %**	**1,9 %**
RAM	**max**	**average**	**current**
	781,3 MB	**763,4 MB**	**780,8 MB**
Modified to 0			
CPU	**max**	**average**	**current**
	1,06 %	**0,99 %**	**1,05 %**
RAM	**max**	**average**	**current**
	781 MB	**746,5 MB**	**780,6 MB**
Default settings			
CPU	**max**	**average**	**current**
	1,20 %	**1,13 %**	**1,16 %**
RAM	**max**	**average**	**current**
	783,6 MB	**758,3 MB**	**782,4 MB**

From the table we can see that by modifying the refresh rate, our CPU and RAM usage did in fact rise, not by much, but bear in mind that we only monitored one device, our Pi must be capable of monitoring dozens, even hundreds of equipments for it to be a true monitoring server.

We can do a bit of math to see how many devices we can actually monitor, the standard CPU and RAM (the Modified to 0) are 0,99 % and 746,5 MB (this is Free RAM, RAM used is 906,5 – 746,5 = 160 MB), the CPU and RAM usage for the default settings are 1,13 % and 918,6 – 758,3 = 160,3 MB, that means an increase of 0,14 % and 0,3 MB for one device and if we like to obtain better results (Modify to 1), the CPU and RAM usage will be 1,87 % and 926,7 – 763,4 = 163,3 MB, that means an increase of 0,88 % and 3,3 MB for one device.

If we go with the modified polling rate we can monitor approximately 113 devices till our CPU reaches 100 %, the RAM usage will enable us to monitor 232 devices till it is used up, so roughly 100 devices can be monitored.

If we go with the standard polling rate we can monitor approximately 708 devices till our CPU reaches 100 %, the RAM usage will enable us to monitor 235 devices till it is used up, so roughly 230 devices can be monitored.

This is theoretical math, the numbers may change, by monitoring more parameters or generating lower traffic, we can only be certain if we test the Pi on a real network, at least we can be sure that it can handle 50 or more devices with a lot of parameters with success.

5. CONCLUSIONS

This paper has achieved its purpose and that is to analyze SNMPv3 on the Raspberry Pi. By describing SNMP, MRTG and the Pi, as well as going into a bit of detail into every one of them, we have successfully managed to document ourselves to implement our design scheme and that involves a computer with Ubuntu installed on it and of course the Pi being on the same network as the PC.

After installing and configuring each element, we are able to test the CPU and RAM usage on the Pi under different tests, under the same traffic load, changing only the refresh rate of the agent so that we can overcome WMI in terms of better and improved resulting data, and thus optimize SNMP.

After setting those O.I.D.'s on the agent side, we have concluded that indeed the CPU and RAM load have risen, a little indeed, but still a noticeable change if we want to monitor more devices with more parameters as resulted from our theoretical math.

Analyzing the results of the tests, we have come to the following conclusion: Raspberry Pi is indeed a powerful monitoring device for its price, and it can be used in the future as a replacement for current expensive servers that occupy a lot of space also.

The next step is to use the Raspberry Pi in a real environment so that we can test our theory and see how many devices can it successfully monitor, maybe even poll more parameters like Temperature, Bandwidth, and so on.

REFERENCES

[1] IETF, A Simple Network Management Protocol, Available 2016, web: (SNMP)https://tools.ietf.org/html/rfc1157

[2] IETF, Simple Network Management Protocol (SNMP) Applications, https://tools.ietf.org/html/rfc3413

[3] Jianguo Ding, Advances in Network Management, 2010 Ed. Taylor and Francis Group, LLC, ISBN 978-1-4200-6452-0

[4] digitalocean, How To Install and Configure an SNMP Daemon and Client on Ubuntu 14.04, Available 2016, web: https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-an-snmp-daemon-and-client-on-ubuntu-14-04

[5] Ubuntu, “Quick HOWTO : Ch23 : Advanced MRTG for Linux”, Available 2016, web: http://wiki.ubuntu.org.cn/Quick_HOWTO_:_Ch23_:_Advanced_MRTG_for_Linux

[6] Tobi Oetiker's MRTG - The Multi Router Traffic Grapher, Available 2016, web: http://oss.oetiker.ch/mrtg