

# **LAB EXPERIMENT: MIB II and SNMP Analysis**

## **LEARNING OBJECTIVES**

Learn how MIB II is structured  
Learn how to obtain management information from the Network components  
Learn how to set up SNMP commands for querying scalar and tabular MIB II information  
Learn how to set up SNMP commands for setting up MIB II information  
Learn to analyze the SNMP Protocol messages using Wireshark  
Learn the differences between the various SNMP Versions: SNMP 1, SNMP2, SNMP3

## **NETWORK CONFIGURATION OF THE EXPERIMENT**

The experiment is based on an educational network (see figure below).

This educational platform is configured so that all the subnet components are accessible from the PC. All routers run the RIPv2 protocol.

The virtual configuration has following components:

- 6 x LAN Subnets: 192.168.40.0/24, 192.168.44.0/24, 192.168.48.0/24, 192.168.50.0/24, 192.168.55.0/24, 192.168.62.0/24
- 4 x Vyatta\*\* Routers: R1, R2, R3, R4;
- 1 x PC

*\*\* Vyatta Router Software is not supported as open source anymore. The successor of Vyatta is the “Vynos” router software. The present experiment will use in the future this new software package for simulating the routers. The features of Vynos routers are very much similar with the ones from Vyatta. Consequently the present exercise instruction manual remains valid.*

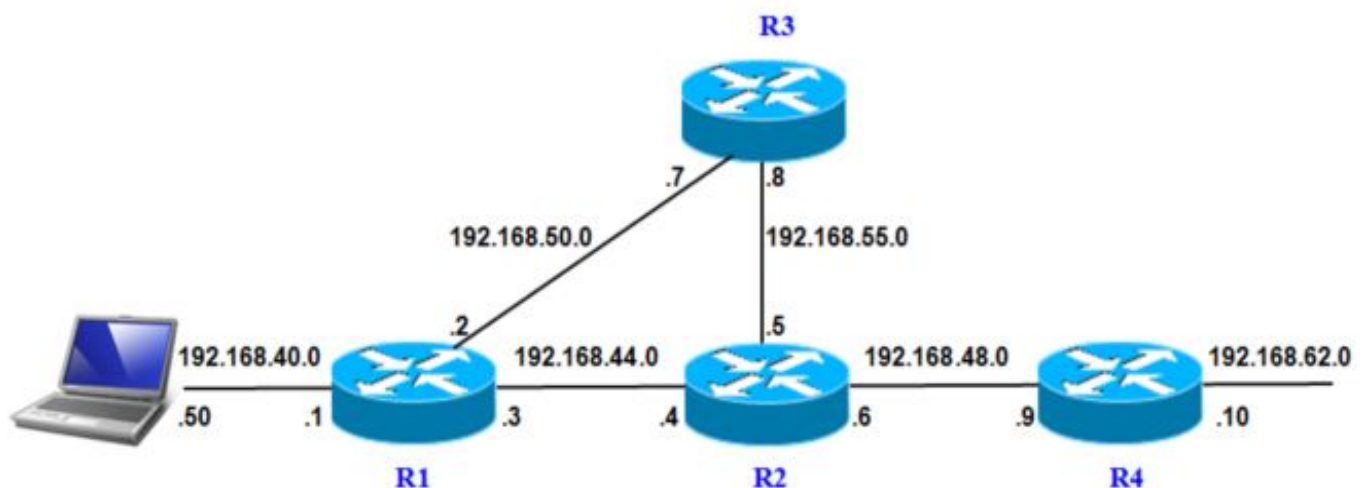


Figure 1 - Network configuration used for this exercise

The routers are installed as software routers running Vyos Router OS.  
Vyos Routers are based on Linux OS.  
Each router runs as a virtual machine within the virtual laboratory.

**Note** *The Lab experiment can be worked through on the basis of a configuration as in Figure 1:*

- *using the installation of the network on vhb-Moodle Platform*
  - *installed by the students on their own computers, or*
  - *installed by the instructor on the lab computers at the university.*
- In this case, the experiment can be carried out using the instruction manual from the virtual laboratory, and the network platform can be installed locally.*
- We refer to this lab procedure as:*  
***“Experiments supported Outside the Virtual Lab”***

### **Vyos SNMP configuration**

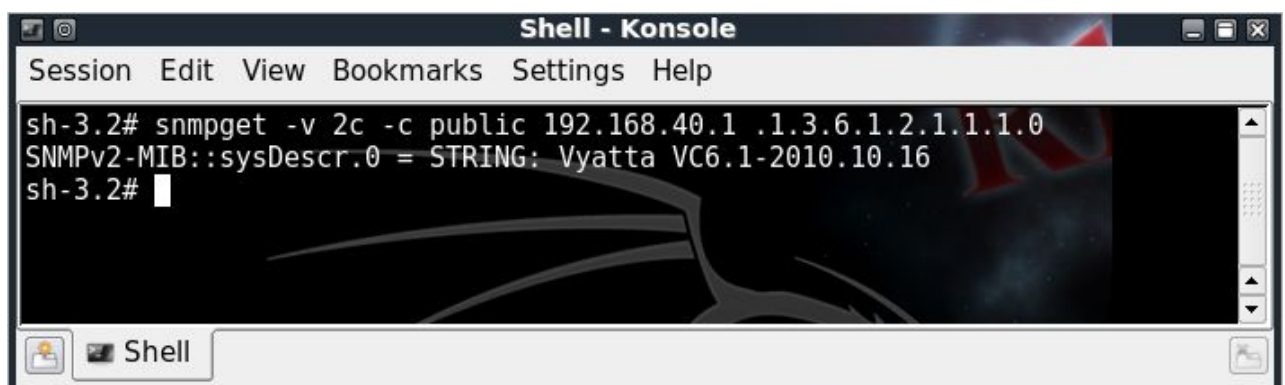
SNMP Agents on the Vyos routers of the network platform are already configured as follows:

- SNMP version: **2c**
- Read-only community string: **public**
- Read-write community string: **private**
- Access is granted to **any IP Address**.

**Note:** *SNMP Version 2c is insecure because the community string is transferred as plain text over the network. It should therefore be used with caution in production networks.*

### **Unix/Linux SNMP commands**

- 1) **snmpget** is used to send SNMP Get Request.
- 2) **snmpgetnext** is used to send SNMP Get-Next Request.
- 3) **snmpwalk** is used to send SNMP Walk Request.
- 4) **snmpset** is used to send SNMP Set Request.



```
Shell - Konsole
Session Edit View Bookmarks Settings Help
sh-3.2# snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.1.1.0
SNMPv2-MIB::sysDescr.0 = STRING: Vyatta VC6.1-2010.10.16
sh-3.2#
```

## Start the network platform

### Step 1:

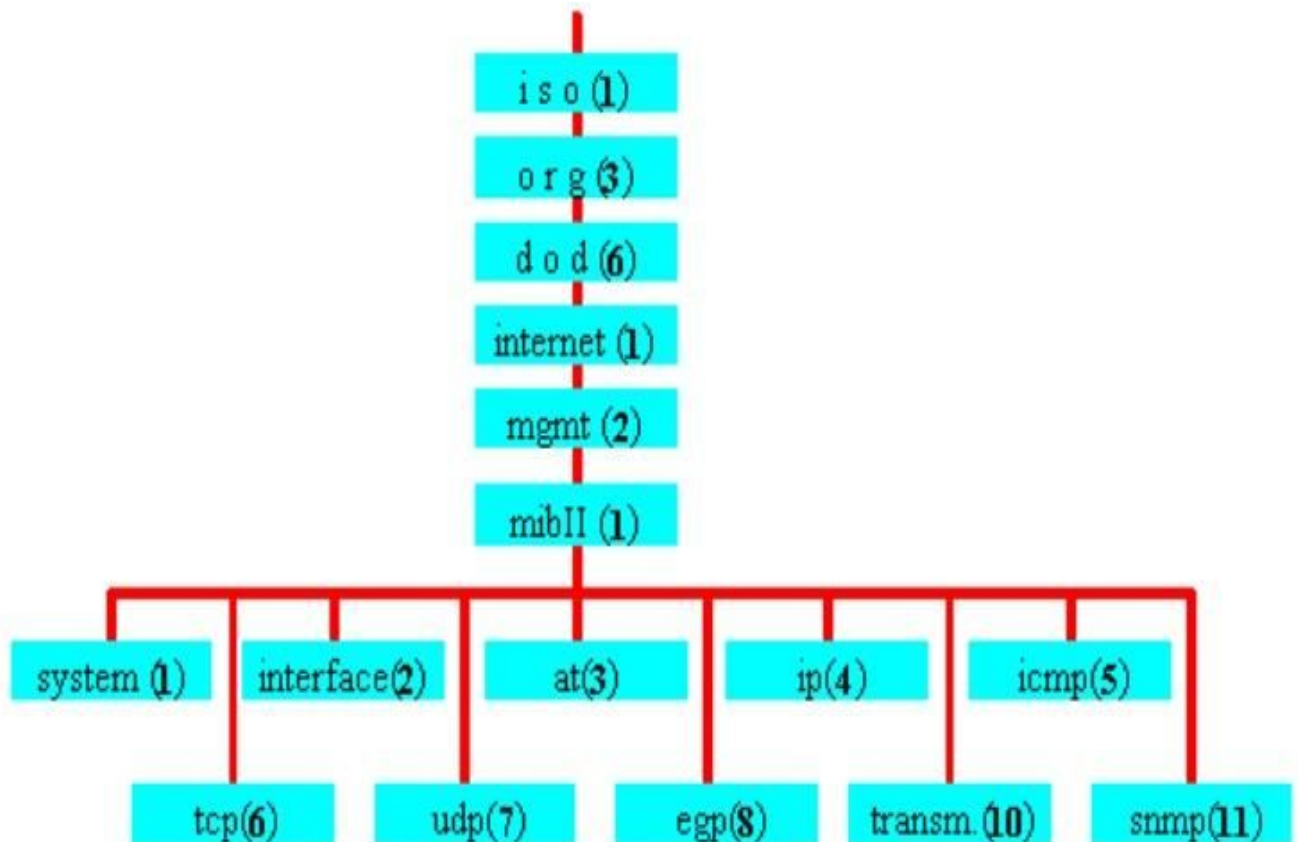
Please start all Virtual Machines:

- 1.1. If the experiment is to be run in the virtual Laboratory (vhb-Moodle Platform), please start all Virtual Machines
- 1.2. If the experiment is to be run *Outside the Virtual Lab*, please ask the instructor where the network platform is located so that you can start all the virtual machines.  
(e.g., in the case of the Munich University of Applied Sciences => please go to: **LAB 16** exercise at **C:\VirtualMachines\Lab16-RIP**)

Note: routing is already preconfigured using RIPv2

### Step 2:

Please open the MIB II tutorial from the Moodle platform in browser of your host PC.



## Accessing scalar objects of the MIB II management bases

### Step 1:

Get System Description information using the *SNMP Get Request* command:

```
snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.1.0
```

Flags/Arguments:

**-v**           => version of SNMP => Vyos is using SNMP version 2c  
**-c**           => community string; **public** is the default setting for RO access  
**192.168.40.1**   => IP address of SNMP agent (Vyos\_R1)  
**.1.3.6.1.2.1.1.0** => OID of System Description (SysDescr)

Please enter the results of the query:

SysDescr = "Vyatta VyOS 1.1.7"

Community string: public

### Step 2:

Get System ObjectID information using the *SNMP Get Next* command

```
snmpgetnext -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.1.0
```

Result: iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.30803

Community string: public

### Step 3:

Get all Objects from the MIB II System Group using the *SNMP Walk Request* command

```
snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.1
```

Result:

```

root@PC1:~# snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.1
iso.3.6.1.2.1.1.1.0 = STRING: "Vyatta VyOS 1.1.7"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.30803
iso.3.6.1.2.1.1.3.0 = Timeticks: (6304426) 17:30:44.26
iso.3.6.1.2.1.1.4.0 = STRING: "root"
iso.3.6.1.2.1.1.5.0 = STRING: "Router1"
iso.3.6.1.2.1.1.6.0 = STRING: "Munich Bavaria"
iso.3.6.1.2.1.1.7.0 = INTEGER: 14
iso.3.6.1.2.1.1.8.0 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.2.1.10.131
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.11 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "RFC 2667 TUNNEL-MIB implementation for Linux 2.2.x kernels."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The management information definitions for the SNMP User-based Security Model"
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "The MIB module for managing UDP implementations"

iso.3.6.1.2.1.1.9.1.3.9 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB modules for managing SNMP Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.11 = STRING: "The MIB module for logging SNMP Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (6) 0:00:00.06
iso.3.6.1.2.1.1.9.1.4.11 = Timeticks: (6) 0:00:00.06

```

Community string: public

#### **Step 4:**

Set System Location information using the *SNMP Set Request* command.

**snmpset -v 2c -c private 192.168.40.1 .1.3.6.1.2.1.1.6.0 s Munich**

Flags/Arguments:

- c       => community string, private is default setting for read-write access
- s       => string => type of record

Other record types:

- i       => Integer = type of record,
- t       => TimeTicks = type of record,
- a       => IPAddress = type of record,
- o       => ObjectID = type of record,
- b       => Bits = type of record,

Result:

```
root@PC1:~# snmpset -v 2c -c private 192.168.40.1 .1.3.6.1.2.1.1.6.0 s Munich
iso.3.6.1.2.1.1.6.0 = STRING: "Munich"
```

Community string: private

#### **Accessing tubular objects of the MIB II management bases**

##### **Step 1:**

Get the IP Address of the Gateway which is used as IP-Next -Hop to send packets to subnet 192.168.62.0.

The information is contained within the *ipRouteTable*

The information can be obtained using the *SNMP Get Request* command:

**snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.21.1.7.192.168.62.0**

Result:

```
root@PC1:~# snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.21.1.7.192.168.62.0
iso.3.6.1.2.1.4.21.1.7.192.168.62.0 = IPAddress: 192.168.44.4
```

##### **Step 2:**

Get first object of the ifTable through the SNMP Get-Next Request command

**snmpgetnext -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.2.2**

Result:

```
root@PC1:~# snmpgetnext -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.2.2
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
```

### **Step 3:**

Get the State of all TCP connections from tcpConnTable using the *SNMP Walk Request* command

```
snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.6.13.1.1
```

Result:

```
root@PC1:~# snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.6.13.1.1
iso.3.6.1.2.1.6.13.1.1.0.0.0.0.80.0.0.0.0.0 = INTEGER: 2
iso.3.6.1.2.1.6.13.1.1.0.0.0.0.443.0.0.0.0.0 = INTEGER: 2
iso.3.6.1.2.1.6.13.1.1.127.0.0.1.199.0.0.0.0.0 = INTEGER: 2
iso.3.6.1.2.1.6.13.1.1.127.0.0.1.199.127.0.0.1.43811 = INTEGER: 5
iso.3.6.1.2.1.6.13.1.1.127.0.0.1.199.127.0.0.1.43814 = INTEGER: 5
iso.3.6.1.2.1.6.13.1.1.127.0.0.1.199.127.0.0.1.43815 = INTEGER: 5
iso.3.6.1.2.1.6.13.1.1.127.0.0.1.43811.127.0.0.1.199 = INTEGER: 5
iso.3.6.1.2.1.6.13.1.1.127.0.0.1.43814.127.0.0.1.199 = INTEGER: 5
iso.3.6.1.2.1.6.13.1.1.127.0.0.1.43815.127.0.0.1.199 = INTEGER: 5
```

### **Step 4:**

Get all data from ipAddrTable at once through the *SNMP Walk Request* command

```
snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.20
```

Result:



```

root@PC1:~# snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.20
iso.3.6.1.2.1.4.20.1.1.127.0.0.1 = IPAddress: 127.0.0.1
iso.3.6.1.2.1.4.20.1.1.192.168.40.1 = IPAddress: 192.168.40.1
iso.3.6.1.2.1.4.20.1.1.192.168.44.3 = IPAddress: 192.168.44.3
iso.3.6.1.2.1.4.20.1.1.192.168.50.2 = IPAddress: 192.168.50.2
iso.3.6.1.2.1.4.20.1.2.127.0.0.1 = INTEGER: 1
iso.3.6.1.2.1.4.20.1.2.192.168.40.1 = INTEGER: 2
iso.3.6.1.2.1.4.20.1.2.192.168.44.3 = INTEGER: 4
iso.3.6.1.2.1.4.20.1.2.192.168.50.2 = INTEGER: 5
iso.3.6.1.2.1.4.20.1.3.127.0.0.1 = IPAddress: 255.0.0.0
iso.3.6.1.2.1.4.20.1.3.192.168.40.1 = IPAddress: 255.255.255.0
iso.3.6.1.2.1.4.20.1.3.192.168.44.3 = IPAddress: 255.255.255.0
iso.3.6.1.2.1.4.20.1.3.192.168.50.2 = IPAddress: 255.255.255.0
iso.3.6.1.2.1.4.20.1.4.127.0.0.1 = INTEGER: 0
iso.3.6.1.2.1.4.20.1.4.192.168.40.1 = INTEGER: 1
iso.3.6.1.2.1.4.20.1.4.192.168.44.3 = INTEGER: 1
iso.3.6.1.2.1.4.20.1.4.192.168.50.2 = INTEGER: 1

```

### **Step 5:**

Set the Metric parameter for route entry within the ipRouteTable to the value = 100

Use the command: *SNMP Walk Request*

```
snmpset -v 2c -c private 192.168.40.1 .1.3.6.1.2.1.4.21.1.3.192.168.62.0 i 100
```

**Note:** Vyos does not support writes on tubular objects, so you will get an error message !!!

## **Using index to retrieve management information base in multiple steps**

### **Step 1:**

Get the MAC Address of the interface through the IP address 192.168.40.1

- 1.1. Get ipAdEntIfIndex of the interface through the IP address 192.168.40.1 from the ipAddrTable

```
snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.20.1.2.192.168.40.1
```

Result: ipAdEntIfIndex =

```

root@PC1:~# snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.20.1.2.192.168.40.1
iso.3.6.1.2.1.4.20.1.2.192.168.40.1 = INTEGER: 2

```

- 1.2. Get ifPhysAdr of the interface through ifIndex  
ifIndex value was obtained during the setup 1.1. (see above)

```
snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.2.2.1.6.<ipAdEntIfIndex>
```



Result: ifPhysAdr =  
**root@PC1**:~# snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.2.2.1.6.2  
iso.3.6.1.2.1.2.2.1.6.2 = Hex-STRING: 08 00 27 83 1D E0

## **Questions**

- 1) Which Transport Protocol and Ports are used for SNMP Protocol ?

UDP, port 161 and 162

- 2) How are the community strings sent: encrypted or not encrypted?

Not Encrypted (Version 3 can encrypt )

- 3) How many reply messages did you capture during the first *SNMP Walk* example?  
Explain your answer.

42. Under the node are 42 leaves.

- 4) Briefly explain the difference between *SNMP Get* and *SNMP GetNext* request.

snmpget gets that specific object

snmpgetnext gets the next node

- 5) Explain: which layer is the SNMP Protocol placed at?

Application Layer.

- 6) Explain: when does an Agent send a TRAP to the Manager?

When there is a failure or significant event, snmp sends the manager a message without a request.

- 7) Indicate the main differences between SNMPv2 and SNMPv3.

Enhanced Security is the primary difference (Authentication and Privacy)

(Authentication of Requests, Encryption of Payloads)

- 8) What are the main differences between MIB II and RMON-MIB ?

RMON has more intelligence on client(devices) site and reduces management overhead. Can sent events to the Network Management System. MIB II only provides a interface to access some information on the switch and need to be requested from the NMS.

- 9) Consider the following SNMP message which was captured using Wireshark analyzer.  
Which Object was requested and which value was returned?  
Explain your answer.

Frame 2 (84 bytes on wire, 84 bytes captured)

Ethernet II, Src: 00:11:92:83:49:20, Dst: 00:06:5b:75:c1:fe

Internet Protocol, Src. Addr: 194.95.109.130 (194.95.109.130), Dst. Addr: 194.95.109.181 (194.95.109.181)

User Datagram Protocol, Src Port: **snmp** (161), Dst Port: 1276 (1276)

Simple Network Management Protocol

Version: 1 (0)

Community: public

PDU type: RESPONSE (2)

Request Id: 0x00000004

Error Status: NO ERROR (0)

Error Index: 0

**Object identifier : ???**

**Value: Counter32: ?????????????????????**

```
0000 00 06 5b 75 c1 fe 00 11 92 83 49 20 08 00 45 00  ..[u.....I ..E.
0010 00 46 90 20 00 00 ff 11 cb 8f c2 5f 6d 82 c2 5f  .F. .... _m._
0020 6d b5 00 a1 04 fc 00 32 08 82 30 28 02 01 00 04  m.....2..0(....
0030 06 70 75 62 6c 69 63 a2 1b 02 01 04 02 01 00 02  .public.....
0040 30 0e 06 08 2b 06 01 02 01 04 04 00 01 00 30 10  ..0.0...+.....
0050 41 02 1c 2b                                     A..+
```

Answer:      OID: 1.3.6.1.2.1.4.4.0 Hex-Value: 41 02 1C 2B

Explanation: \_\_\_\_\_

- 10) Consider the SNMP Listing No. 1 and 2.  
The "Labserver" host from the Laboratory was queried using an SNMP browser.

10.1. Please identify what kind of interfaces are connected to the Labserver?

Please explain this task again in the praktikum.

10.2. Please identify what kind of IP addresses are assigned to these interfaces?

Please explain this task again in the praktikum.

### **Listing 1**

#### **C:\>snmputil walk labserver public 1.3.6.1.2.1.4.20.1.2**

```
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.127.0.0.1
Value = INTEGER - 1
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.192.168.0.140
Value = INTEGER - 16777219
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.192.168.10.140
Value = INTEGER - 16777219
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.192.168.20.140
Value = INTEGER - 16777219
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.192.168.30.140
Value = INTEGER - 16777219
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.194.95.109.66
Value = INTEGER - 16777220
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.194.95.109.82
Value = INTEGER - 16777220
Variable = ip.ipAddrTable.ipAddrEntry.ipAdEntIfIndex.194.95.109.140
Value = INTEGER - 16777221
End of MIB subtree.
```

#### **C:\>snmputil walk labserver public 1.3.6.1.2.1.2.2.1.2**

```
Variable = interfaces.ifTable.ifEntry.ifDescr.1
Value = OCTET STRING - MS TCP Loopback interface<0x0>
Variable = interfaces.ifTable.ifEntry.ifDescr.16777219
Value = OCTET STRING - Intel(R) PRO/100+ Server Adapter (PILA8470B)<0x0>
Variable = interfaces.ifTable.ifEntry.ifDescr.16777220
Value = OCTET STRING - Intel(R) PRO/100+ Server Adapter (PILA8470B)<0x0>
Variable = interfaces.ifTable.ifEntry.ifDescr.16777221
Value = OCTET STRING - Intel(R) PRO/100 Network Connection<0x0>
End of MIB subtree.
```

### **Listing 2**

#### **C:\>snmputil walk labserver public 1.3.6.1.2.1.2.2.1.3**

```
Variable = interfaces.ifTable.ifEntry.ifType.1
Value = INTEGER - 24
Variable = interfaces.ifTable.ifEntry.ifType.16777219
Value = INTEGER - 6
Variable = interfaces.ifTable.ifEntry.ifType.16777220
Value = INTEGER - 6
Variable = interfaces.ifTable.ifEntry.ifType.16777221
Value = INTEGER - 6
End of MIB subtree.
```

#### **C:\>snmputil walk labserver public 1.3.6.1.2.1.2.2.1.5**

```
Variable = interfaces.ifTable.ifEntry.ifSpeed.1
Value = Gauge - 10000000
Variable = interfaces.ifTable.ifEntry.ifSpeed.16777219
Value = Gauge - 100000000
Variable = interfaces.ifTable.ifEntry.ifSpeed.16777220
Value = Gauge - 10000000
Variable = interfaces.ifTable.ifEntry.ifSpeed.16777221
```

Value = Gauge - 100000000  
End of MIB subtree.

## **Lab Results**

Please fill out the following tables with your lab results and submit it for evaluation to your Moodle platform.

Note: All experiments are carried out using the lab16\_rip starting configuration.

<b>Accessing scalar objects of the MIB II management information base</b>	
STEP1: snmpget -v 2c -c public 192.168.40.1 .1.3.6.2.1.1.1.0	SysDescr:  Community String:
STEP2: snmpgetnext -v 2c -c public 192.168.40.1 .1.3.6.2.1.1.1.0	Result:  Community String:
STEP3: snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.2.1.1	Result:  Community String:
STEP4: snmpset -v 2c -c private 192.168.40.1 .1.3.6.1.2.1.1.6.0 s Munich	Result:  Community String:
<b>Accessing tabular objects of the MIB II management information base</b>	
<b>Queries</b>	<b>Results</b>
STEP1: snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.21.1.7.192.168.62.0	Result:
STEP2: snmpgetnext -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.2.2	Result:
STEP3: snmpwalk -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.6.13.1.1	Result:

STEP4: snmpwalk -v 2c -c public 19.168.40.1 .1.3.6.1.2.1.4.20	Result:
STEP5: snmpset -v 2c -c private 192.168.40.1 .1.3.6.1.2.1.4.21.1.3.192.168.62.0 i 100	Result:
<b>Using index to retrieve management information base in multiple steps</b>	
STEP1: snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.4.20.1.2.192.168.40.1	Result:
STEP2: snmpget -v 2c -c public 192.168.40.1 .1.3.6.1.2.1.2.2.1.6.2	Result:



<b>Questions</b>
ANSWER 1:
ANSWER 2:
ANSWER 3:
ANSWER 4:
ANSWER 5:
ANSWER 6:
ANSWER 7:
ANSWER 8:
ANSWER 9: Object Identifier: VALUE:  Explanation:
ANSWER 10: 10.1:  10.2: