Simulation of an SNMP Agent: Operations, Analysis and Results

Pradeep Kumar Sharma [1], Dr. S.S Tyagi [2]
[1 2] Department of Computer Science & Engineering
[1] Mtech, CSE, MRIU, Faridabad, Haryana, India
[2] Prof. and HOD CSE MRIU, Faridabad, Haryana, India
[1] Pradeep.hi86@gmail.com

Abstract- **This paper aims at evaluate an SNMP environment on the basis of its operations and practical approaches. The SNMP protocol is used to monitor, control and configuring Network elements. Even though the SNMP technology is well documented but still it is relatively unclear how the SNMP is used in real practice. This paper discusses about how the SNMP is deployed in a real network and how the traffic is analyzed and controlled with the help of SNMP. With the continued improvements in the performance of the SNMP data collection, the developers of the SNMP based network monitoring system are applying their best effort in the system development. SNMP network management system development is an important aspect of the network management, and development process requires a lots of coordination with network environment, but it is too costly to construct a real network for development of network management system, so if we can provide a simulation network environment to develop a network management system, it will bring a great convenience for testing, training and other aspects of SNMP .**

Keywords - SNMP, ADVENT Net, MIB, SMI.

I. INTRODUCTION

In today's complex network of routers, switches, and servers, it can seem like a daunting task to manage all the devices on your network and make sure they're not only up and running but performing optimally. This is where the Simple Network Management Protocol (SNMP) can help. SNMP was introduced in 1988 to meet the growing need for a standard for managing Internet Protocol (IP) devices. SNMP provides its users with a "simple" set of operations that allows these devices to be managed remotely. The core of SNMP is a simple set of operations (and the Information these operations gather) that gives administrators the ability to change the state of some SNMP-based device. For example, you can use SNMP to shut down an interface on your router or check the speed at which your Ethernet interface is operating. SNMP can even monitor the temperature on your switch and warn you when it is too high.

The rest of the paper is organized as follows: in part II we will discuss about the different versions of SNMP,in part III we describe how SNMP works, in part IV we will simulate an SNMP environment and its operations, part V features the discussion of our results and we will conclude the analysis in part VI.

II. SNMP VERSIONS

SNMP Version 1 (SNMPv1) is the standard version of the SNMP protocol. It's defined in RFC 1157 and is a full IETF standard. SNMPv1's security is based on communities, which are nothing more than passwords: plain-text strings that allow any SNMP-based application that knows the strings to gain access to a device's management information. There are typically three communities in SNMPv1: read-only, read-write, and trap. SNMP Version 2 (SNMPv2) is the next version of SNMP issued in 1996, with the functional enhancement but without a security facility. This version uses a community based security feature as in SNMPv1 and is known as SNMPv2c. SNMP Version 3(SNMPv3) is not a standalone replacement of the SNMPv1and/or SNMPv2, We can define the SNMPv3 as SNMPv1 and/or SNMPv2 with security mechanism, authentication, privacy and access control.

III. SNMP WORKING

Before describing the working of SNMP we will discuss the elements of an SNMP environment. An SNMP management system contains:

- Several nodes each having an SNMP entity which contains a command responder and notification originator applications. These nodes are called Agents.
- At least one SNMP entity containing command generator and/or notification receiver applications. This is called a manager.
- A management protocol which is used to convey information between these two SNMP entities.
- A database which reflect the status of a physical link and resides in the Agent. This is called MIB.

The basic working of SNMP includes the reading of (OIDs) Object Identifiers implemented in the MIBs and provides the related information to the Manager. The Manager does this by using several monitoring and managing operations. The monitoring operations are performed by the manager by sending the commands GET, GETNEXT, GETBULK and answered by the agents. On the basis of this the manager collect the information about the different devices in the network based on the inbuilt MIB in the Agent. A manager can also perform a managing act by sending a SET request to modify agent parameters and agent responds to manager with the status of set command. Multiple requests can also be send by a single SNMP packet by using multiple OIDs in GET request or by using a GETBULK command for a range of variables. Moreover the data can also be send by agent side independently without any request by manager in the form of TRAP messages.

IV. SIMULATION of SNMP AGENT

There are many papers which discuss about the SNMP and its functionality, operations but nobody has incorporated about the practical environment how these operations takes place and how the data is retrieved from devices and analyzed at the manager side. This paper basically creates the practical approach towards the SNMP as how the operations work. We have used an adventnet SNMP simulator for representing all the activities related to SNMP.

A. Description of Simulator

AdventNet Simulation Toolkit is a comprehensive and versatile set of intuitive tools that simplify development, testing, and demonstration of network management products without requiring real devices. The SNMP Agent Simulator, bundled with Simulation Toolkit, is a platform-independent JAVA-based tool, which is used to simulate SNMPv1, SNMPv2C and SNMPv3 manageable devices. You can run this tool on Windows / Linux / Solaris and simulate one agent at a time. The SNMP Agent Simulator takes the MIB file as input and simulates instance variables for the MIBs. These instance variables are saved into a file and the agent responds from this file. You can create default variables from the MIB or learn variables from an existing agent.

B. Steps of simulation

The SNMP Agent Simulator is used to simulate a Standalone SNMP Agent and is purely SNMP specific. It supports SNMPv1/v2c/v3 versions. Now, let us walk through the steps involved in simulating a standalone SNMP Agent

1. Specifying the Input to the SNMP Agent Simulator:

   The first step in simulating an SNMP agent, is to load the MIB file or an existing Configuration file in the SNMP Agent Simulator. When you have completed loading the MIBs and simulating default values, a basic SNMP simulation is created.
2. Configuring Values:

   Any variation on this basic simulation can be created for use with the SNMP Agent Simulator. You can configure scalar and table values, record real agents, populate the SNMP table, configure traps and informs, configure real time behavior of devices using Jython scripts, simulate error conditions and scenarios and so on. In case of SNMPv3, the v3 configuration tool is used to configure SNMP v3 users.
3. Testing and Editing Tools

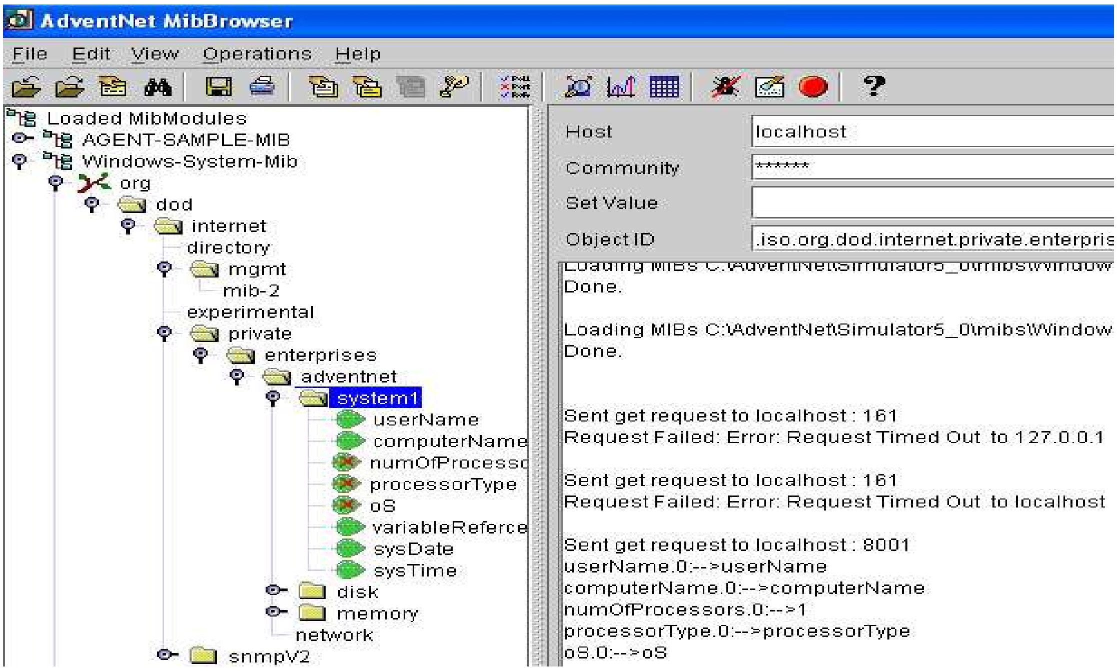
   AdventNet Simulation Toolkit is packaged along with Test and Editor tools to provide the user with the complete Simulation experience.

- Test Tools serve as Manager Applications to test the simulated SNMP agent.
    - MIB Browser : To test the simulated SNMPv1/v2/v3 agent.
- Editor Tools are used to edit some of the input files of the simulator.
    - SNMPv3 Config Tool : To configure user entries in USM and VACM tables for the SNMPv3 agent.

C. SNMP Operations

For the simulation of SNMP operations first of all we will have to load an MIB file into the SNMP agent simulator and we will simulate it for the default values. Now for the manager related activites we will have to introduced MIB browser which works as an SNMP Manager.

SNMP GET:
SNMP get is used to retrieve the information about the network variables from the inbuilt MIB in the agent.



Host: localhost
Community: ******
Set Value:
Object ID: .iso.org.dod.internet.private.enterpris

```
Loading MIBs C:\AdventNet\Simulator5_0\mibs\Window
Done.

Loading MIBs C:\AdventNet\Simulator5_0\mibs\Window
Done.


Sent get request to localhost : 161
Request Failed: Error: Request Timed Out  to 127.0.0.1

Sent get request to localhost : 161
Request Failed: Error: Request Timed Out  to localhost

Sent get request to localhost : 8001
userName.0:-->userName
computerName.0:-->computerName
numOfProcessors.0:-->1
processorType.0:-->processorType
oS.0:-->oS
```

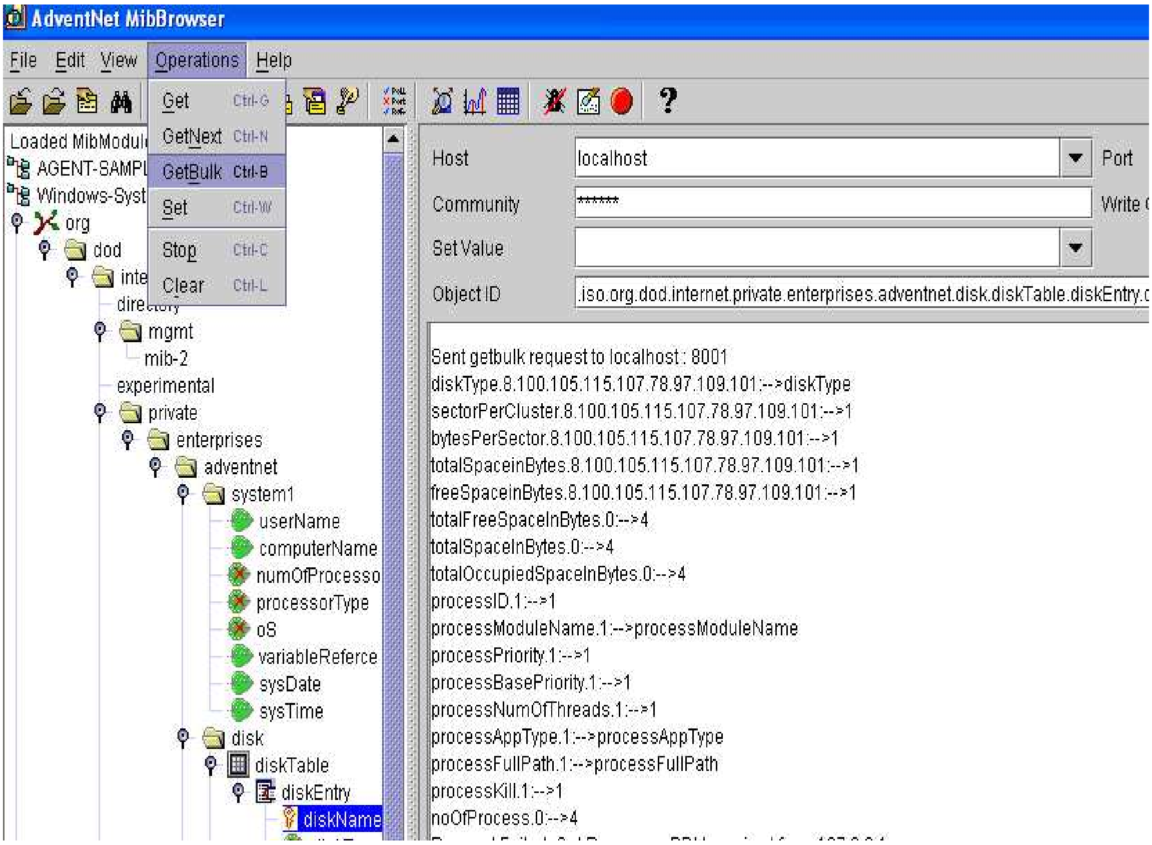
Fig.1 SNMP GET Operation

The GET operation is explained in the figure 1. In the left side of the figure, we have loaded a Windows-system-mib and by expanding the MIB tree we have performed the GET operation for the System1 network element, and in the right bottom side of the figure we have the data of all variables of system1.

SNMP GETNEXT:
SNMP getnext is used to retrive the information about the next variable. This command is useful when we want retrieves the values one by one.

SNMP GETBULK:

SNMP getbulk command is used in SNMPv2 and SNMPv3 for retrieving more variables in a single request. Since it is a Version 2 and 3 command so we have to tell the simulator to use a version 2 PDU for this operation. In this operation the agent responds with as much as it can send in a single request. The standard get operation can attempt to retrieve more than one MIB object at once, but message sizes are limited by the agent's capabilities. If the agent can't return all the requested responses, it returns an error message with no data The get-bulk operation, on the other hand, tells the agent to send as much of the response back as it can. This means that incomplete responses are possible. Two fields must be set when issuing a get-bulk command: nonrepeaters and max-repetitions. Nonrepeaters tells the get-bulk command that the first N objects can be retrieved with a simple get-next operation. Max-repetitions tells the get-bulk command to attempt up to M get-next operations to retrieve the remaining objects.



Host: localhost

Community: ******

Set Value:

Object ID: .iso.org.dod.internet.private.enterprises.adventnet.disk.diskTable.diskEntry.d

```
Sent getbulk request to localhost : 8001
diskType.8.100.105.115.107.78.97.109.101:-->diskType
sectorPerCluster.8.100.105.115.107.78.97.109.101:-->1
bytesPerSector.8.100.105.115.107.78.97.109.101:-->1
totalSpaceinBytes.8.100.105.115.107.78.97.109.101:-->1
freeSpaceinBytes.8.100.105.115.107.78.97.109.101:-->1
totalFreeSpaceInBytes.0:-->4
totalSpaceInBytes.0:-->4
totalOccupiedSpaceInBytes.0:-->4
processID.1:-->1
processModuleName.1:-->processModuleName
processPriority.1:-->1
processBasePriority.1:-->1
processNumOfThreads.1:-->1
processAppType.1:-->processAppType
processFullPath.1:-->processFullPath
processKill.1:-->1
noOfProcess.0:-->4
```

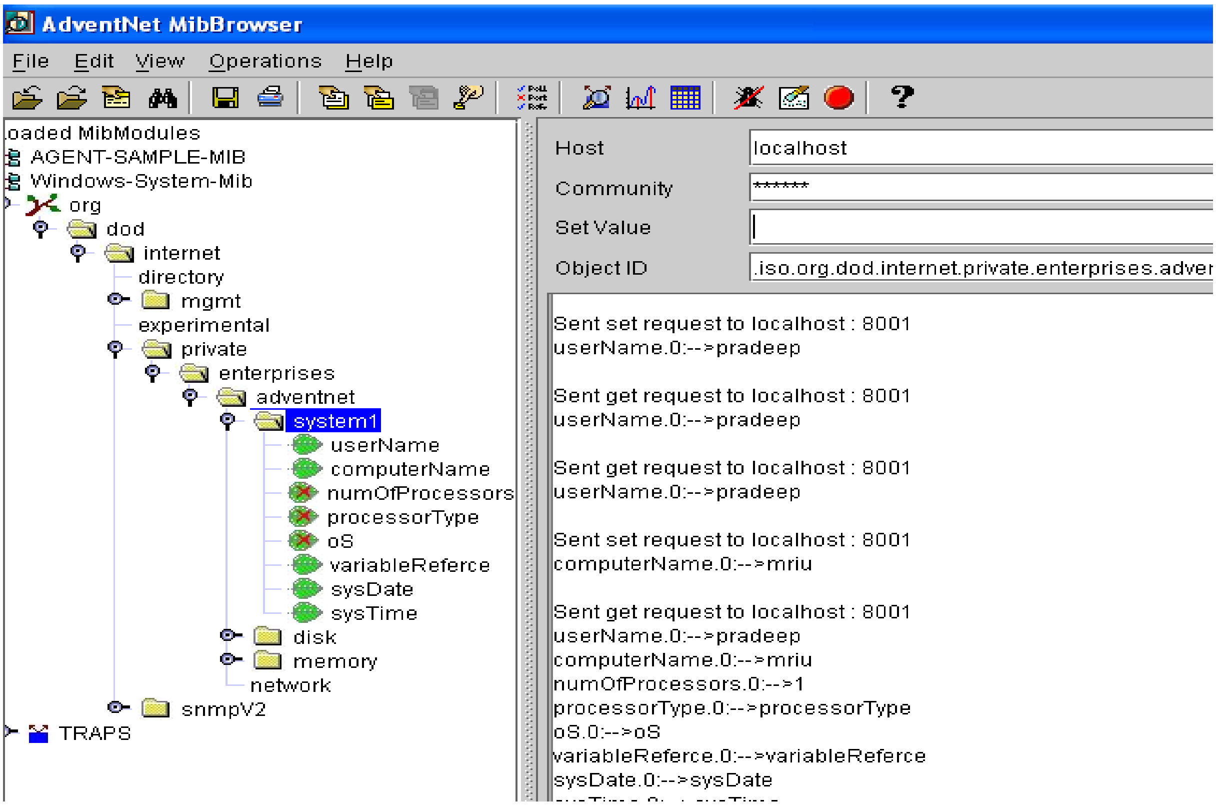
Figure.2 The GETBULK Operation

As shown in figure getbulk operation has retrived a lots of variables in single request.

SNMP SET:

The set command is used to change the value of a managed object or to create a new row in a table. Objects that are defined in the MIB as read-write or write-only can be altered or created using this command. It is possible for an NMS to set more than one object at a time. By using this command we are managing the network with help of

SNMP, we can change the status of a network variable.



Host: localhost
Community: ******
Set Value:
Object ID: .iso.org.dod.internet.private.enterprises.adver

```
Sent set request to localhost : 8001
userName.0:-->pradeep

Sent get request to localhost : 8001
userName.0:-->pradeep

Sent get request to localhost : 8001
userName.0:-->pradeep

Sent set request to localhost : 8001
computerName.0:-->mriu

Sent get request to localhost : 8001
userName.0:-->pradeep
computerName.0:-->mriu
numOfProcessors.0:-->1
processorType.0:-->processorType
oS.0:-->oS
variableReference.0:-->variableReference
sysDate.0:-->sysDate
```

Fig. 3. SNMP SET Operation

As shown in the fig. 3 the SET command has been introduced for changing the value Username and Computername from 0 to pradeep and mriu respectivly. After setting the values it has again checked by GET request.

SNMP TRAP

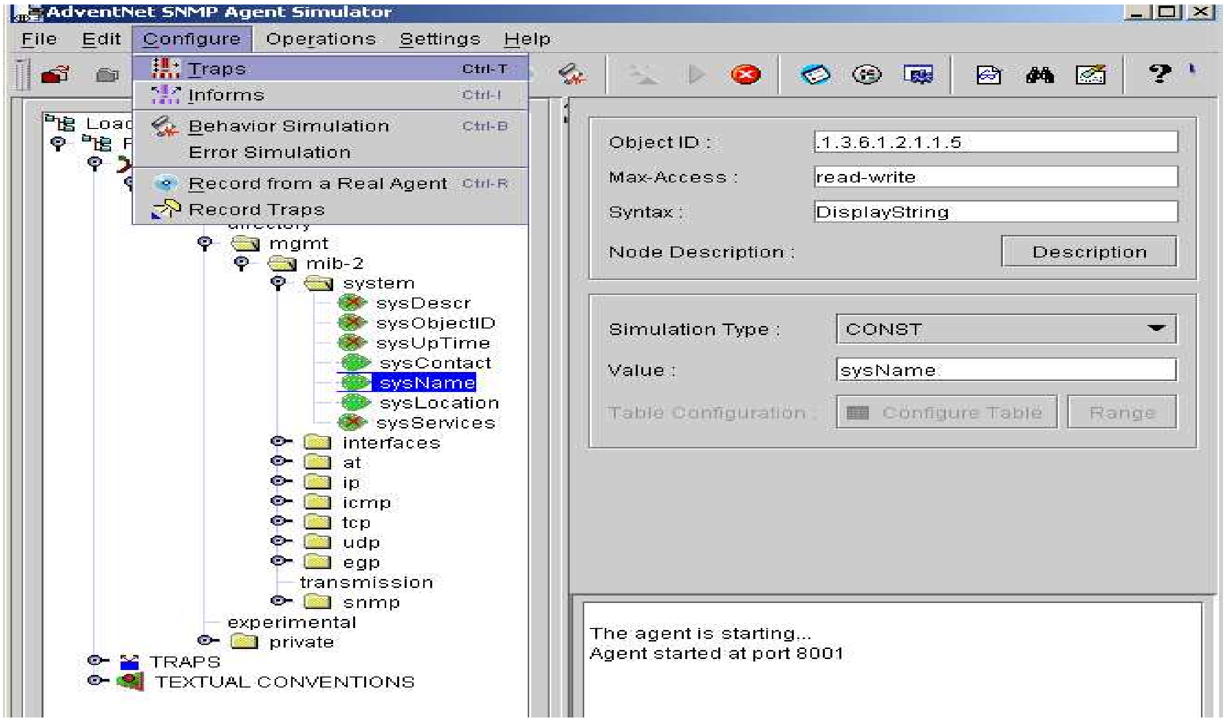
The agent, when faced by some problem or error in the transmission of message, responds to the manager by sending unsolicited messages called traps.

Traps are unsolicited messages sent from an SNMP agent to one or more SNMP Management applications. It is an asynchronous notification sent by the agent to the manager about some event occurrence in the device. Trap messages are received at port 162.

For the simulation of Trap messages, first we have to configure traps in the SNMP Agent simulator. The traps can be generated on following conditions.

- Request Based Trap
- Threshold Based Trap
- Timer Based trap

In the request base, trap occurs when a request (get, set) is made by manager, in threshold based we match a threshold OID value and condition, and for Time based we specify a specific time.



Object ID : .1.3.6.1.2.1.1.5
Max-Access : read-write
Syntax : DisplayString
Simulation Type : CONST
Value : sysName

```
The agent is starting...
Agent started at port 8001
```

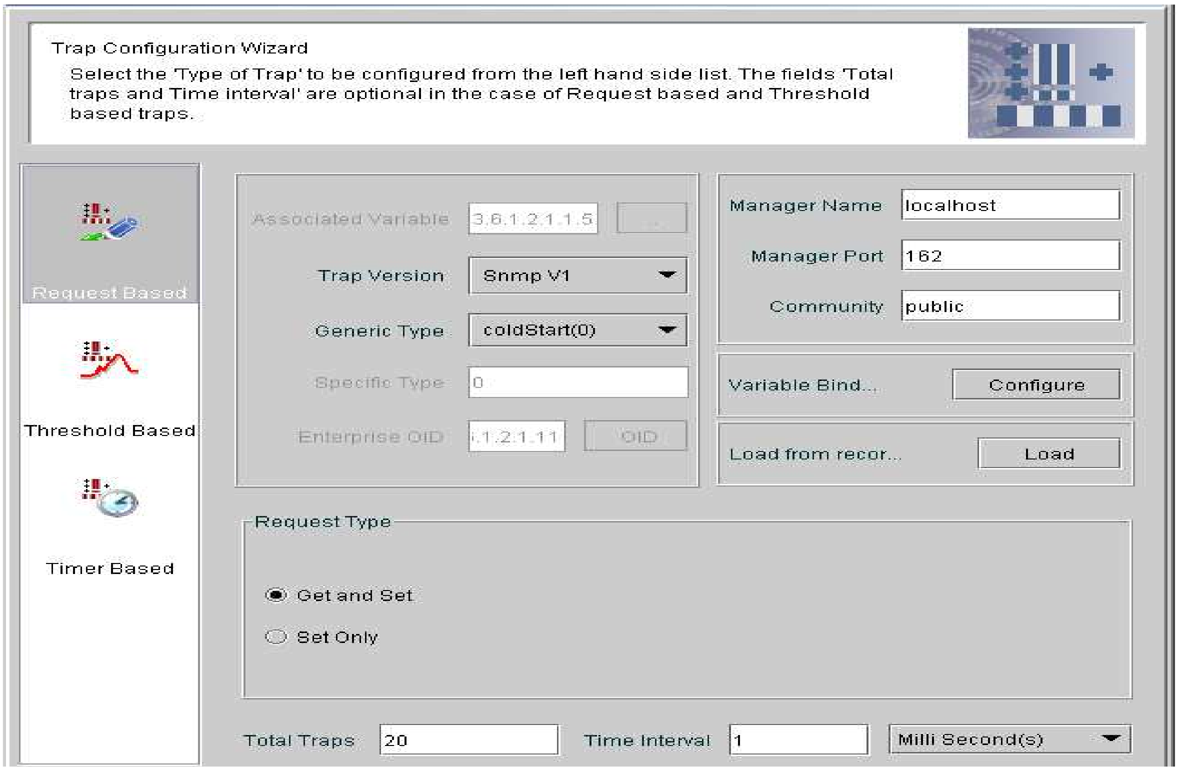
Fig. 4.1 Configuring traps

We are configuring a request based trap in the figure. 4.1, 4.2 , 4.3 and 4.4.

In figure 4.1. a trap is to be configured for the variable sysName. As shown in figure 4.2 it is a request based trap, whenever a Get or Set request will be send for the variable sysName a trap should occur. Here we have configured 20 traps with an interval of 1 milisecond.

For representing the traps we use Trap Viewer which is the part of the MIB browser. In the figure 4.3 we are performing the get and set operations for variable sysName and we see the traps in the trap viewer as shown in figure 4.4.

When an NMS receives a trap, it needs to know how to interpret it; that is, it needs to know what the trap means and how to interpret the information it carries. A trap is first identified by its generic trap number. There are even generic trap numbers (0-6). Generic trap 6 is a special atch-all category for "enterprise-specific" traps, which are traps defined by vendors or users that fall outside of the six generic trap categories. Enterprise-specific traps are further identified by an enterprise ID (i.e., an object ID somewhere in the enterprises branch of the MIB tree, iso.org.dod.internet.private.enterprises) and a specific trap number chosen by the enterprise that defined the trap.



Trap Configuration Wizard

Select the 'Type of Trap' to be configured from the left hand side list. The fields 'Total traps and Time interval' are optional in the case of Request based and Threshold based traps.

Associated Variable: 3.6.1.2.1.1.5

Trap Version: Snmp V1

Generic Type: coldStart(0)

Specific Type: 0

Enterprise OID: .1.2.1.11

Manager Name: localhost

Manager Port: 162

Community: public

Request Type

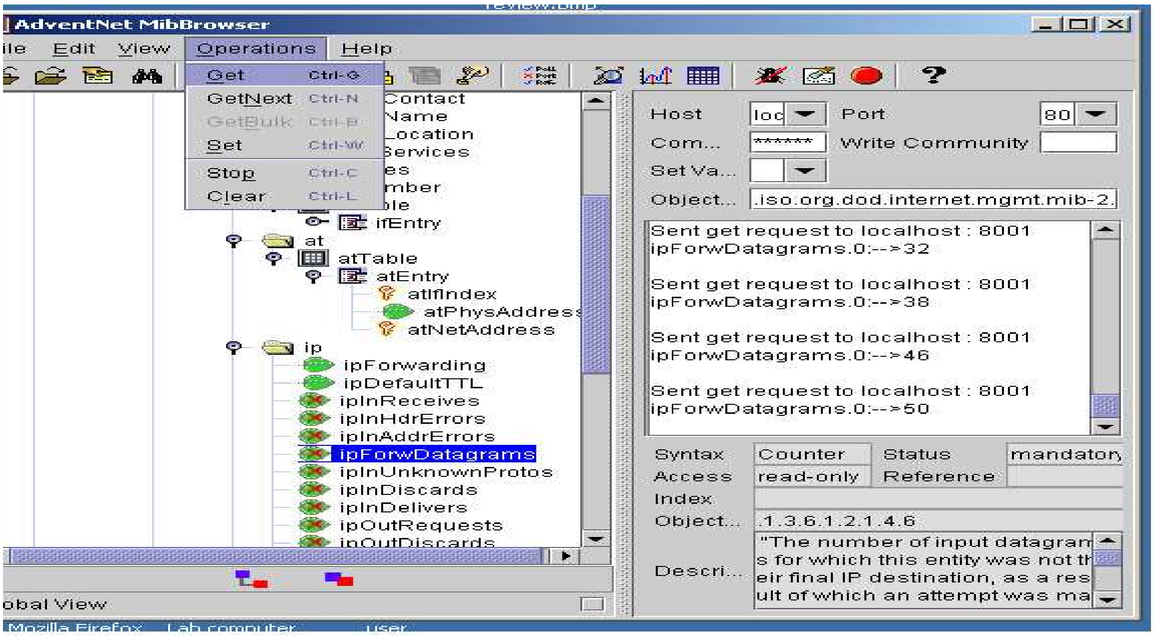
● Get and Set

○ Set Only

Total Traps: 20

Time Interval: 1 Milli Second(s)

Fig. 4.2



AdventNet MibBrowser

Host: loc Port: 80

Com...: ****** Write Community:

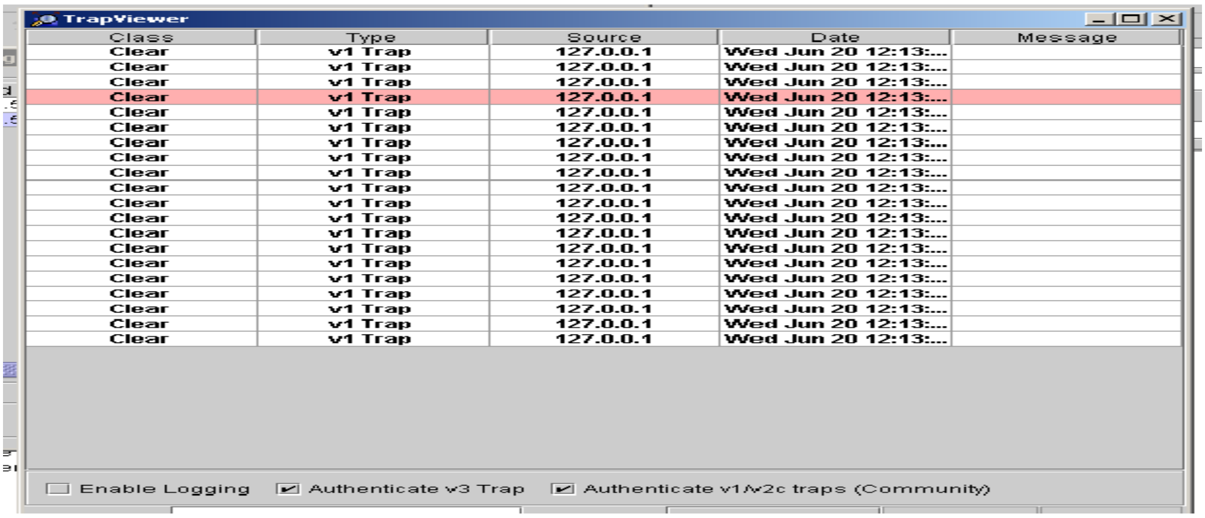
Set Va...:

Object...: .iso.org.dod.internet.mgmt.mib-2.

```
Sent get request to localhost : 8001
ipForwDatagrams.0:-->32

Sent get request to localhost : 8001
ipForwDatagrams.0:-->38

Sent get request to localhost : 8001
ipForwDatagrams.0:-->46

Sent get request to localhost : 8001
ipForwDatagrams.0:-->50
```

Syntax	Counter	Status	mandatory
Access	read-only	Reference	
Index			
Object...	.1.3.6.1.2.1.4.6		
Descri...	"The number of input datagram s for which this entity was not th eir final IP destination, as a res ult of which an attempt was ma		

Fig. 4.3



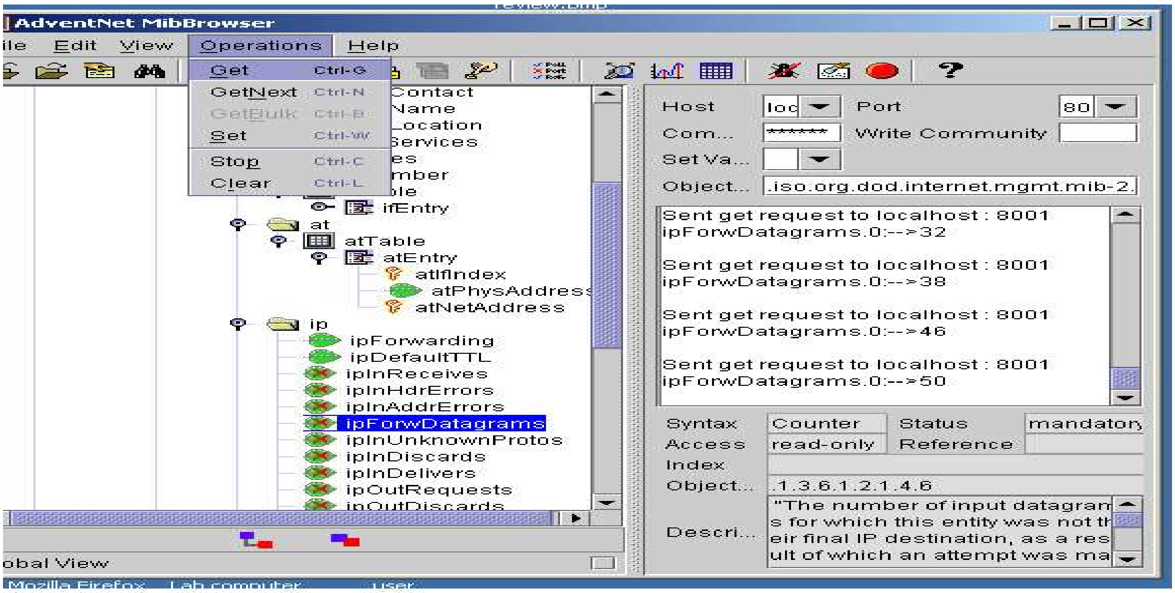
Class	Type	Source	Date	Message
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	
Clear	v1 Trap	127.0.0.1	Wed Jun 20 12:13:...	

□ Enable Logging ☑ Authenticate v3 Trap ☑ Authenticate v1/v2c traps (Community)

Fig. 4.4

SNMP Graph:

We can also graph the SNMP information for better understanding the current traffic and its behavior. In the Figure 5 we are performing the get operation for the variable ipForwDatagrams and graph with respect to this operation is shown in figure 6.

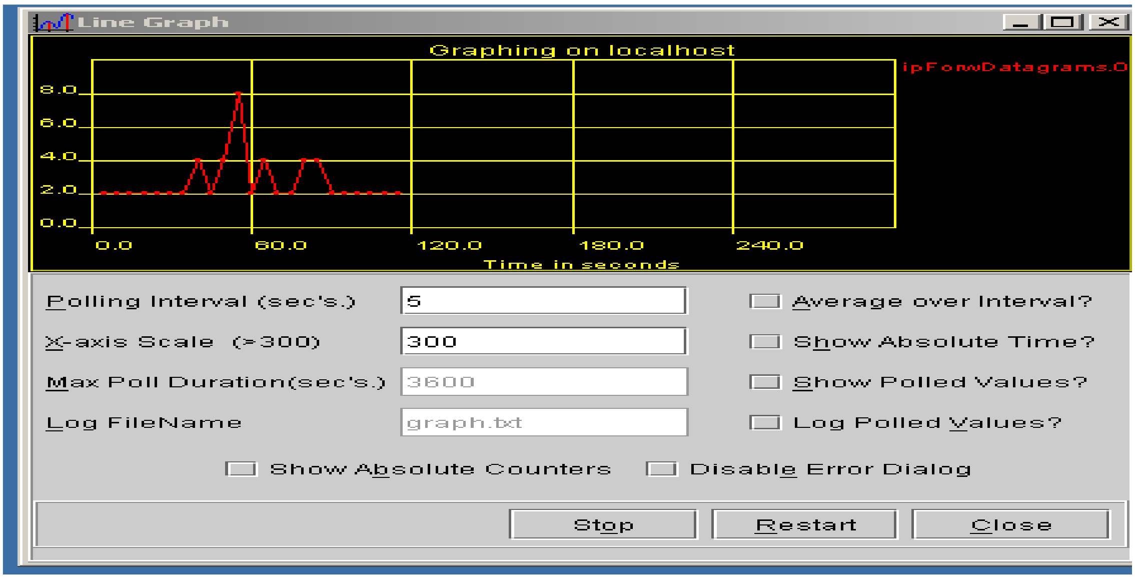


Host loc Port 80
Com... ******* Write Community
Set Va...
Object... .iso.org.dod.internet.mgmt.mib-2.

```
Sent get request to localhost : 8001
ipForwDatagrams.0:-->32

Sent get request to localhost : 8001
ipForwDatagrams.0:-->38

Sent get request to localhost : 8001
ipForwDatagrams.0:-->46

Sent get request to localhost : 8001
ipForwDatagrams.0:-->50
```

Syntax	Counter	Status	mandatory
Access	read-only	Reference	
Index			
Object...	.1.3.6.1.2.1.4.6		
Descri...	"The number of input datagrams for which this entity was not their final IP destination, as a result of which an attempt was ma		

Fig 5.



Polling Interval (sec's.) 5
X-axis Scale (>300) 300
Max Poll Duration(sec's.) 3600
Log FileName graph.txt

□ Average over Interval?
□ Show Absolute Time?
□ Show Polled Values?
□ Log Polled Values?
□ Show Absolute Counters
□ Disable Error Dialog

Figure 6. Graphing on localhost

V. SIMULATION RESULTS

By this simulation we have got to know that how the different operations takes place in an SNMP environment. We have seen the difference between get and getnext operation. We have also demonstrated that how to edit the value of a network variable by using the set operation. Trap messages, from the agents to the manager has also been explained. For visualizing the performance of SNMP at different operations in a scenario has also been exhibited.

VI. CONCLUSIONS

To solve the problem of lack of testing environment in the development of SNMP management system, we have simulated the SNMP environment and its operations. The simulation results show that how the manager and agents works together for managing the network. This paper is also useful for the network managers as well as other researchers interested in network management in general. The future work is to improve the process of simulation and adding more and more variables in simulation process and showing the more results.

References

[1] "SNMP-Simple Network Management Protocol" available at http://www.snmplink.org/

[2] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A simple network management protocol (SNMP)" RFC 1157, IETF, 1990. Wesley, 1999

[3] Matthias Wiesmann, Peter Urban, Xavier Defago, "An SNMP based failure detection service", 25th IEEE Symposium on Reliable Distributed Systems (SRDS'06)

[4] AdventNet MIB Browser Available at http://www.adventnet.com/products/snmputilities/mib-browser.html

[5] Abstract Syntax Notation One Available at http://asn1.elibel.tm.fr/

[6] W. Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2," Addison

[7] B. Pagurek, Y. Wang, and T. White, "Integration of Mobile Agents with SNMP: Why and How," IEEE/IFI Network Operations and Management Symposium, 2000.

[8] Periklis Chatzimisios, Security issues and vulnerabilities of the SNMP protocol, 1st International Conference on Electrical and Electronics Engineering, 2004.

[9] Ethereal: An opensource network sniffer available at http://www.ethereal.com/

[10] HP OpenView Network Services Management solutions available at http://www.openview.hp.com/solutions/

[11]Advent Net Manage Engine OpUtils: A Comprehensive Network Monitoring Toolset available at http://manageengine.adventnet.com/products/oputils/index.html.