MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

IV SEMESTER B.TECH. (CCE)

ICT 2226 Computer Network Protocols

CNP Group Assignment-FISAC 2

Analysis of Network Packet and its Protocols using Wireshark

Date: 05-04-2025

Group no – 4

230953140 – Mayurika Sathish
230953142 – Sruthi D V
230953170 – Atharv Hrishikesh Pawar
230953172 – Yashraj Sandeep Sakunde

Protocol:

SNMP

Overview

This report presents a comprehensive analysis of the Simple Network Management Protocol (SNMP) through simulation and packet tracing using MIB browser and Wireshark. The study encompasses the generation and capture of SNMP packets, an exploration of the protocol's structure and functionality, and a detailed examination of packet flows and hierarchies.

Objectives

- Configure an SNMP agent and manager to communicate, and use Wireshark to capture the exchanged packets.
- Analyse SNMP's structure, including versions, message types, and the roles of agents, managers, and the MIB.
- Utilize Wireshark's I/O Graphs to visualize SNMP packet flow and assess network performance.
- Examine SNMP's encapsulation within other protocols using Wireshark's protocol hierarchy feature.

Key Components:

1. MIB Browser
2. Wireshark
3. SNMP agent

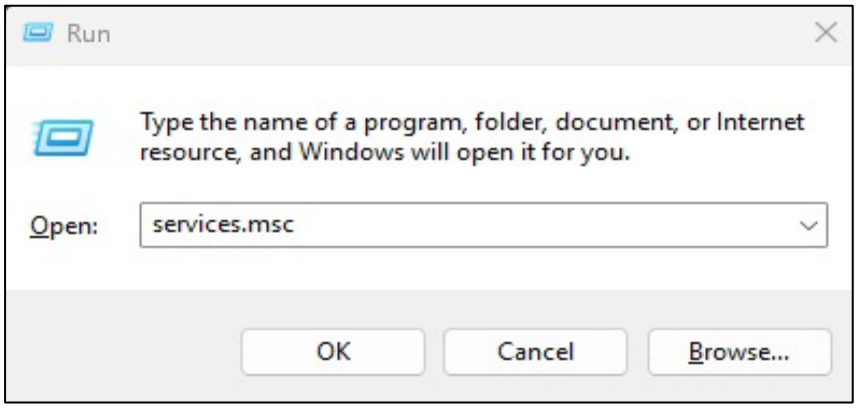
Procedure:

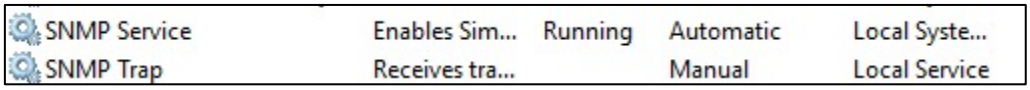
Following are the steps for generation and capturing of SNMP Packets using MIB browser:

1) Installing Wireshark and setting up Wireshark

a) First, install wireshark from https://www.wireshark.org, making sure to include the Npcap packet capture driver during installation. Next, install and configure the SNMP service on the windows machine. This is done by going to Settings → Apps → Optional features → Add a feature, then searching for and installing "Simple Network Management Protocol (SNMP)". After installation, open the Services window by typing services.msc in the Run dialog, find "SNMP Service", right-click it and select Properties. In the Security tab, add a Read-only community string such as public and optionally configure the allowed IP addresses that can query the SNMP agent.

To determine the IP address of the system (which will act as the SNMP Agent), open Command Prompt, type ipconfig, and press Enter. Note the IPv4 address shown under the active network adapter (such as Wi-Fi or Ethernet). This IP will be used in the MIB Browser to connect to the SNMP Agent.





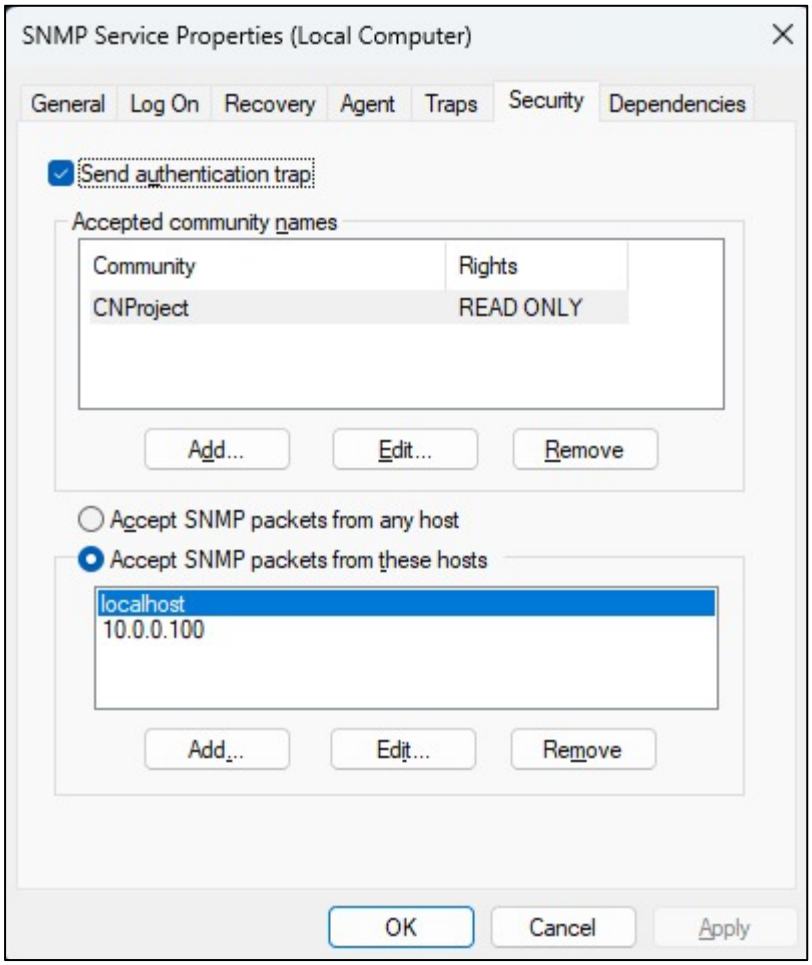


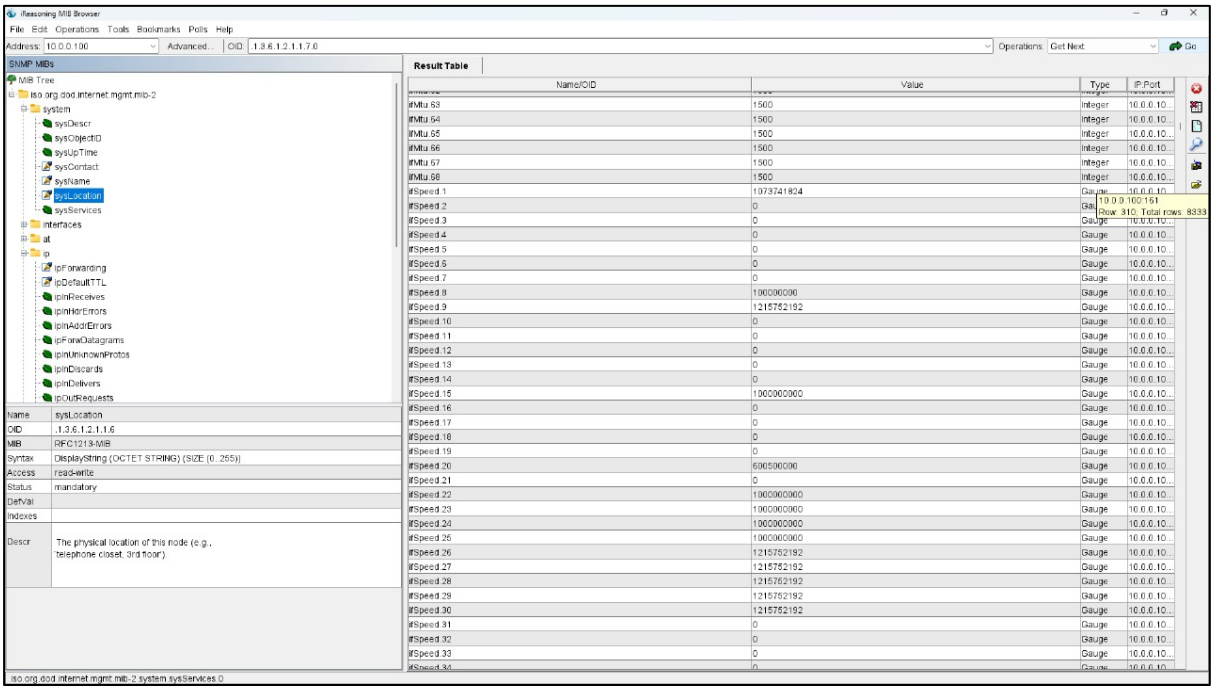
Fig. Configuration on local computer to allow exchange of SNMP packets

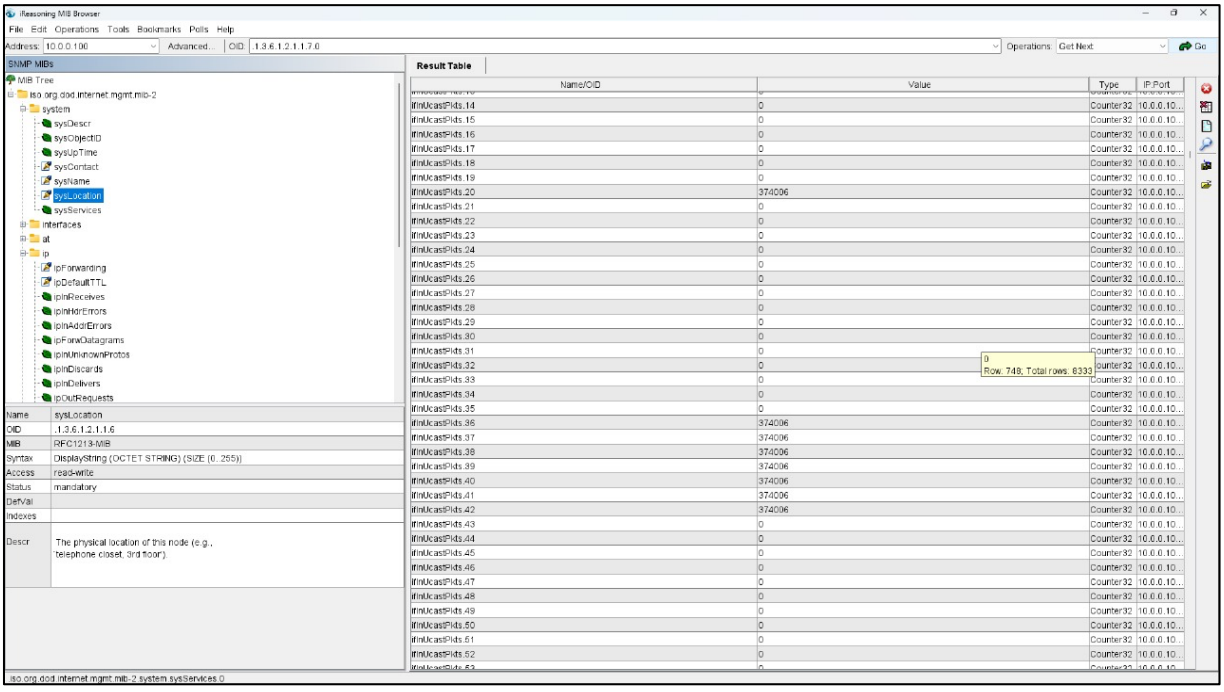
2) Installing and utilising MIB browser to perform required actions:

Next, open iReasoning's MIB Browser. The MIB Browser functions as the SNMP Manager, which is responsible for sending SNMP requests to SNMP Agents and receiving responses. In the Address field of the MIB Browser, enter the IP address of the SNMP Agent, and in the Community field, enter the read community string that was configured earlier (e.g. public)

Now open Wireshark, select the appropriate network interface (such as Wi-Fi or Ethernet) and click the start button to begin capturing packets. In the filter bar, type snmp to focus on SNMP protocol traffic.

To generate SNMP traffic, use the MIB Browser to perform actions such as GET, GET NEXT, WALK, SET or GET BULK. These operations allow the SNMP Manager to request information or perform queries on the SNMP Agent's Management Information Base (MIB). For example, a GET request retrieves the value of a specific object, while a WALK performs a sequence of GET NEXT operations to traverse the MIB tree. Each of these actions generates SNMP packets that are transmitted to the agent and can be seen in Wireshark in real-time.


iReasoning MIB Browser
File Edit Operations Tools Bookmarks Polls Help
Address: 10.0.0.100
Advanced...
OID: .1.3.6.1.2.1.1.7.0
Operations: Get Next
Go
SNMP MIBs
MIB Tree
iso.org.dod.internet.mgmt.mib-2
system
sysDescr
sysObjectID
sysUpTime
sysContact
sysName
sysLocation
sysServices
interfaces
at
ip
ipForwarding
ipDefaultTTL
ipInReceives
ipInHdrErrors
ipInAddrErrors
ipForwDatagrams
ipInUnknownProtos
ipInDiscards
ipInDelivers
ipOutRequests
Result Table
Name/OID
Value
Type
IP:Port
ifMtu.63 1500 Integer 10.0.0.10...
ifMtu.64 1500 Integer 10.0.0.10...
ifMtu.65 1500 Integer 10.0.0.10...
ifMtu.66 1500 Integer 10.0.0.10...
ifMtu.67 1500 Integer 10.0.0.10...
ifMtu.68 1500 Integer 10.0.0.10...
ifSpeed.1 1073741824 Gauge 10.0.0.10...
ifSpeed.2 0 Gauge 10.0.0.10...
ifSpeed.3 0 Gauge 10.0.0.10...
ifSpeed.4 0 Gauge 10.0.0.10...
ifSpeed.5 0 Gauge 10.0.0.10...
ifSpeed.6 0 Gauge 10.0.0.10...
ifSpeed.7 0 Gauge 10.0.0.10...
ifSpeed.8 100000000 Gauge 10.0.0.10...
ifSpeed.9 1215752192 Gauge 10.0.0.10...
ifSpeed.10 0 Gauge 10.0.0.10...
ifSpeed.11 0 Gauge 10.0.0.10...
ifSpeed.12 0 Gauge 10.0.0.10...
ifSpeed.13 0 Gauge 10.0.0.10...
ifSpeed.14 0 Gauge 10.0.0.10...
ifSpeed.15 1000000000 Gauge 10.0.0.10...
ifSpeed.16 0 Gauge 10.0.0.10...
ifSpeed.17 0 Gauge 10.0.0.10...
ifSpeed.18 0 Gauge 10.0.0.10...
ifSpeed.19 0 Gauge 10.0.0.10...
ifSpeed.20 600500000 Gauge 10.0.0.10...
ifSpeed.21 0 Gauge 10.0.0.10...
ifSpeed.22 1000000000 Gauge 10.0.0.10...
ifSpeed.23 1000000000 Gauge 10.0.0.10...
ifSpeed.24 1000000000 Gauge 10.0.0.10...
ifSpeed.25 1000000000 Gauge 10.0.0.10...
ifSpeed.26 1215752192 Gauge 10.0.0.10...
ifSpeed.27 1215752192 Gauge 10.0.0.10...
ifSpeed.28 1215752192 Gauge 10.0.0.10...
ifSpeed.29 1215752192 Gauge 10.0.0.10...
ifSpeed.30 1215752192 Gauge 10.0.0.10...
ifSpeed.31 0 Gauge 10.0.0.10...
ifSpeed.32 0 Gauge 10.0.0.10...
ifSpeed.33 0 Gauge 10.0.0.10...
10.0.0.102:161
Row: 310, Total rows: 8333
Name sysLocation
OID .1.3.6.1.2.1.1.6
MIB RFC1213-MIB
Syntax DisplayString (OCTET STRING) (SIZE (0..255))
Access read-write
Status mandatory
DefVal
Indexes
Descr The physical location of this node (e.g., 'telephone closet, 3rd floor').
iso.org.dod.internet.mgmt.mib-2.system.sysServices.0


iReasoning MIB Browser
File Edit Operations Tools Bookmarks Polls Help
Address: 10.0.0.100
Advanced...
OID: .1.3.6.1.2.1.1.7.0
Operations: Get Next
Go
SNMP MIBs
MIB Tree
iso.org.dod.internet.mgmt.mib-2
system
sysDescr
sysObjectID
sysUpTime
sysContact
sysName
sysLocation
sysServices
interfaces
at
ip
ipForwarding
ipDefaultTTL
ipInReceives
ipInHdrErrors
ipInAddrErrors
ipForwDatagrams
ipInUnknownProtos
ipInDiscards
ipInDelivers
ipOutRequests
Result Table
Name/OID
Value
Type
IP:Port
ifInUcastPkts.14 0 Counter32 10.0.0.10...
ifInUcastPkts.15 0 Counter32 10.0.0.10...
ifInUcastPkts.16 0 Counter32 10.0.0.10...
ifInUcastPkts.17 0 Counter32 10.0.0.10...
ifInUcastPkts.18 0 Counter32 10.0.0.10...
ifInUcastPkts.19 0 Counter32 10.0.0.10...
ifInUcastPkts.20 374006 Counter32 10.0.0.10...
ifInUcastPkts.21 0 Counter32 10.0.0.10...
ifInUcastPkts.22 0 Counter32 10.0.0.10...
ifInUcastPkts.23 0 Counter32 10.0.0.10...
ifInUcastPkts.24 0 Counter32 10.0.0.10...
ifInUcastPkts.25 0 Counter32 10.0.0.10...
ifInUcastPkts.26 0 Counter32 10.0.0.10...
ifInUcastPkts.27 0 Counter32 10.0.0.10...
ifInUcastPkts.28 0 Counter32 10.0.0.10...
ifInUcastPkts.29 0 Counter32 10.0.0.10...
ifInUcastPkts.30 0 Counter32 10.0.0.10...
ifInUcastPkts.31 0 Counter32 10.0.0.10...
ifInUcastPkts.32 0 Counter32 10.0.0.10...
ifInUcastPkts.33 0 Counter32 10.0.0.10...
ifInUcastPkts.34 0 Counter32 10.0.0.10...
ifInUcastPkts.35 0 Counter32 10.0.0.10...
ifInUcastPkts.36 374006 Counter32 10.0.0.10...
ifInUcastPkts.37 374006 Counter32 10.0.0.10...
ifInUcastPkts.38 374006 Counter32 10.0.0.10...
ifInUcastPkts.39 374006 Counter32 10.0.0.10...
ifInUcastPkts.40 374006 Counter32 10.0.0.10...
ifInUcastPkts.41 374006 Counter32 10.0.0.10...
ifInUcastPkts.42 374006 Counter32 10.0.0.10...
ifInUcastPkts.43 0 Counter32 10.0.0.10...
ifInUcastPkts.44 0 Counter32 10.0.0.10...
ifInUcastPkts.45 0 Counter32 10.0.0.10...
ifInUcastPkts.46 0 Counter32 10.0.0.10...
ifInUcastPkts.47 0 Counter32 10.0.0.10...
ifInUcastPkts.48 0 Counter32 10.0.0.10...
ifInUcastPkts.49 0 Counter32 10.0.0.10...
ifInUcastPkts.50 0 Counter32 10.0.0.10...
ifInUcastPkts.51 0 Counter32 10.0.0.10...
ifInUcastPkts.52 0 Counter32 10.0.0.10...
0
Row: 748, Total rows: 8333
Name sysLocation
OID .1.3.6.1.2.1.1.6
MIB RFC1213-MIB
Syntax DisplayString (OCTET STRING) (SIZE (0..255))
Access read-write
Status mandatory
DefVal
Indexes
Descr The physical location of this node (e.g., 'telephone closet, 3rd floor').
iso.org.dod.internet.mgmt.mib-2.system.sysServices.0

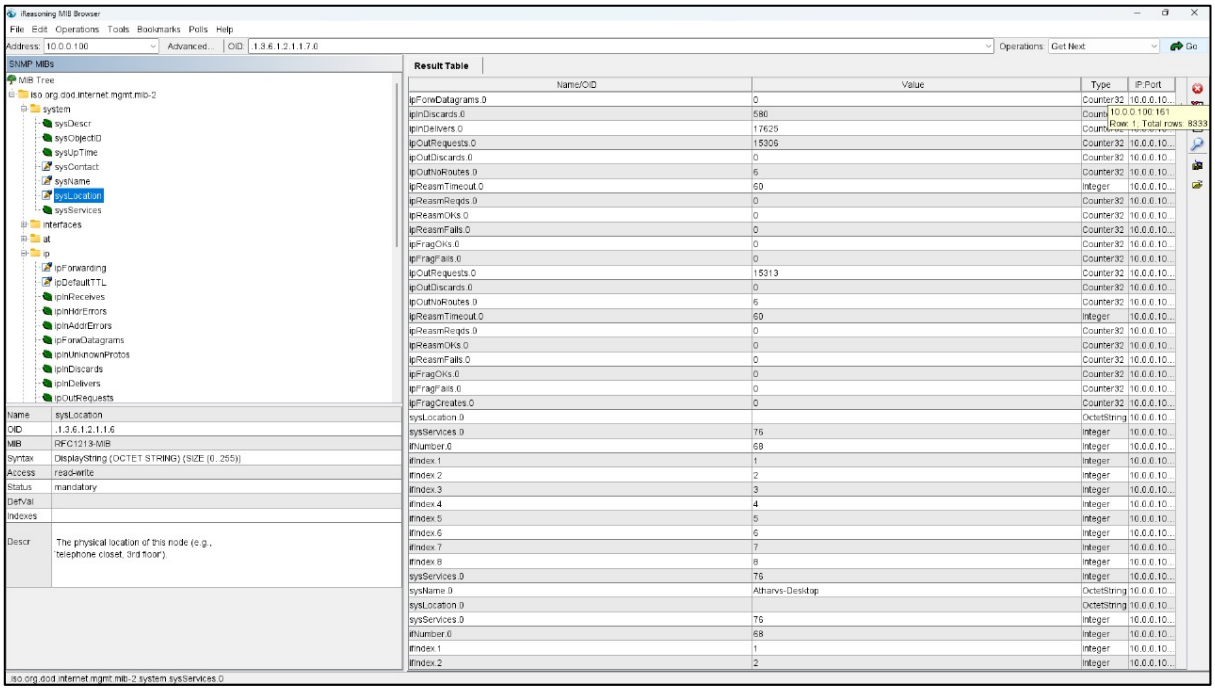


Fig. Configuring MIB Browser to managing and monitoring agents

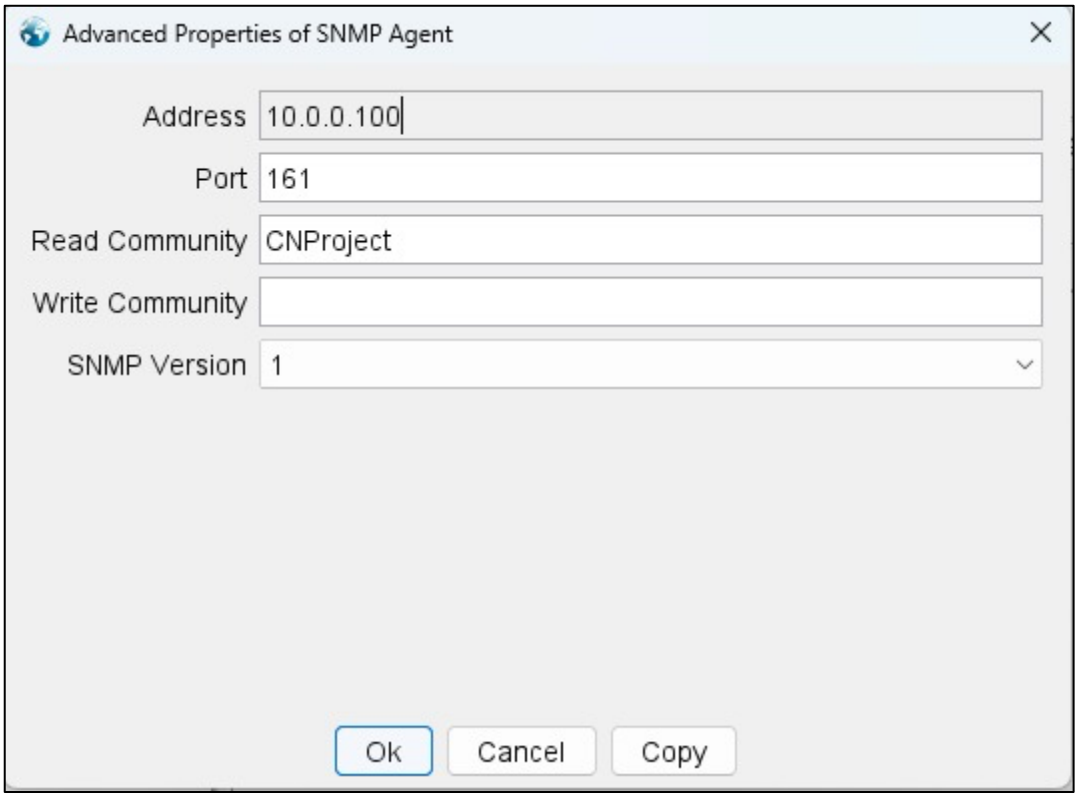


Fig. Setting up SNMP Agent

Once the desired SNMP interactions have been completed, click Stop in Wireshark to end the packet capture. You can then analyze individual SNMP messages, examine packet headers and data, and save the capture using File > Save As for documentation or further analysis.

3) Capturing SNMP packets in Wireshark

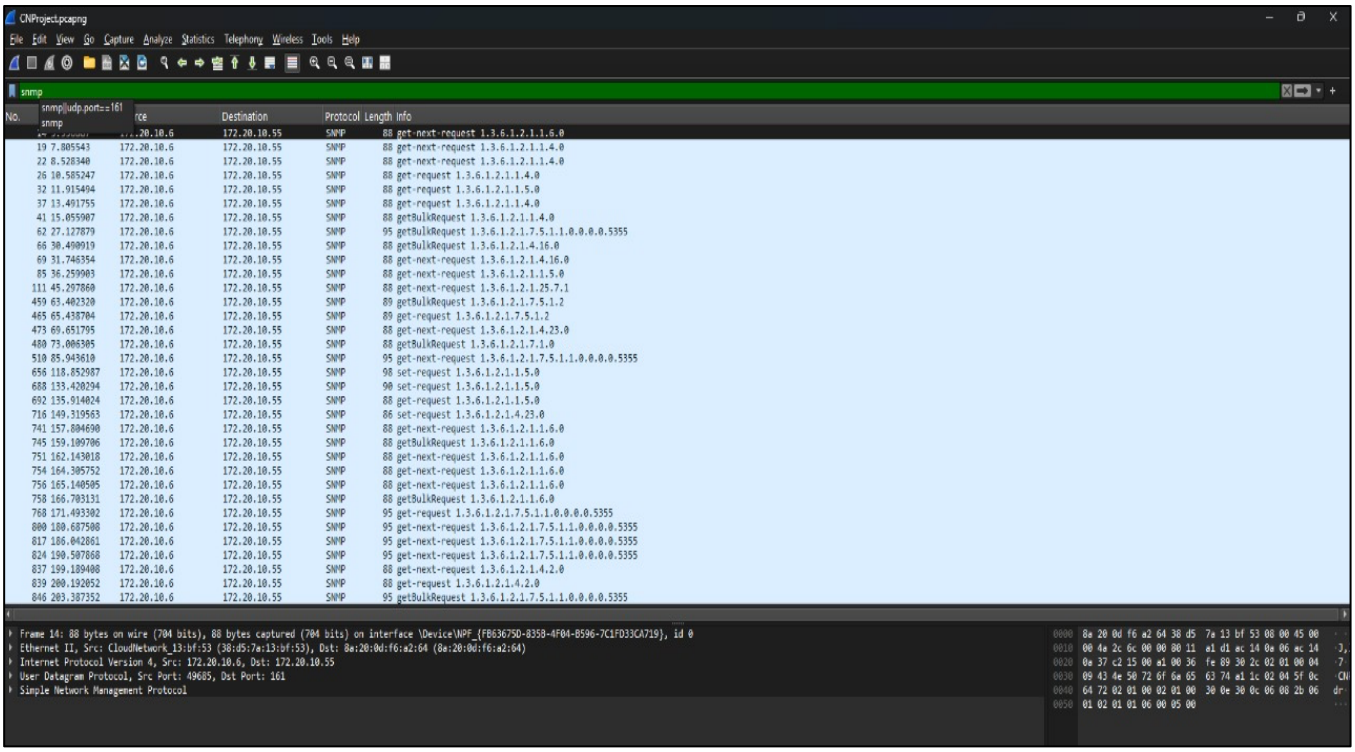


Fig. SNMP packets being captured in Wireshark following the commands *get-next-request, get-request, getBulkRequest and set-request*

SNMP Protocol:

Simple Network Management Protocol (SNMP) is a standard protocol used to monitor and manage network devices like routers, switches, and servers. It works on the application layer and enables communication between an SNMP manager and agents running on devices.

key-components:

- Manager: Requests information or sends commands
- Agent: Responds with data from the device
- MIB (Management Information Base): A database of device parameters.

Flow (I/O) Graph and it's Analysis:

c) Show the flow (I/O) graph & try to analyse the flow.

The first IO graph shows the full network capture with generally low traffic & occassional spikes, around 40-60s & 100-120s, likely due to SNMP activity. The TCP errors are present but minimal. This graph provides an overview of all network traffic during the session.

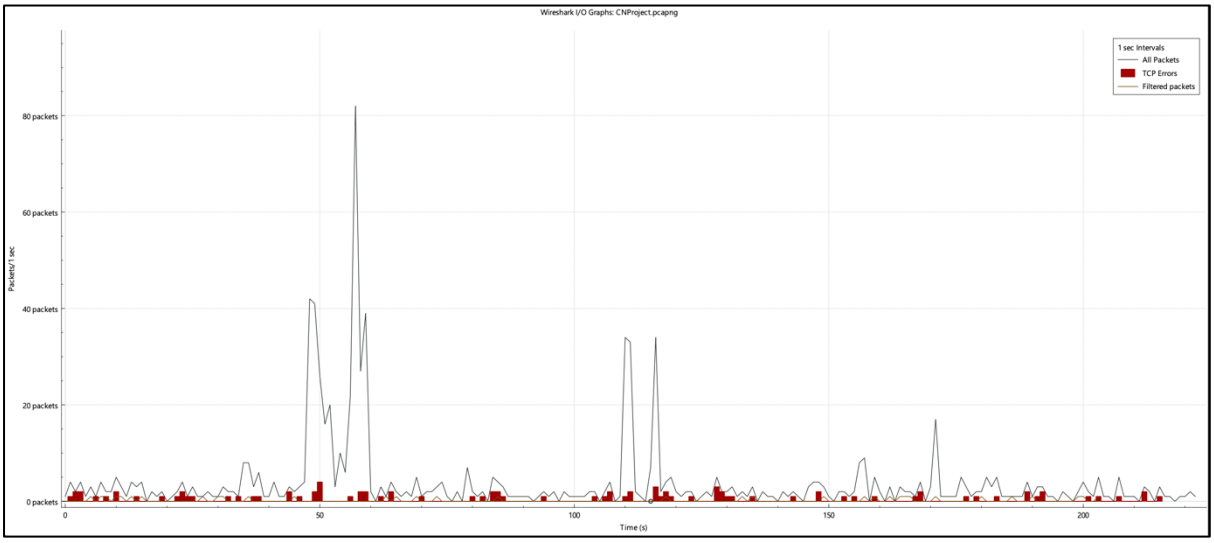


Fig. I/O graph demonstrating all protocol packets in the network over time

The 2nd graph is filtered to show only SNMP packets. It displays sharp, irregular spikes indicating short bursts of SNMP traffic, typical of operations like GET, WALK & BULK GET. These bursts reflect active communication b/w the SNMP manager (MIB browser) & agent.

IN CONCLUSION:
The pattern suggests that SNMP traffic was generated in focused intervals rather than continuously.

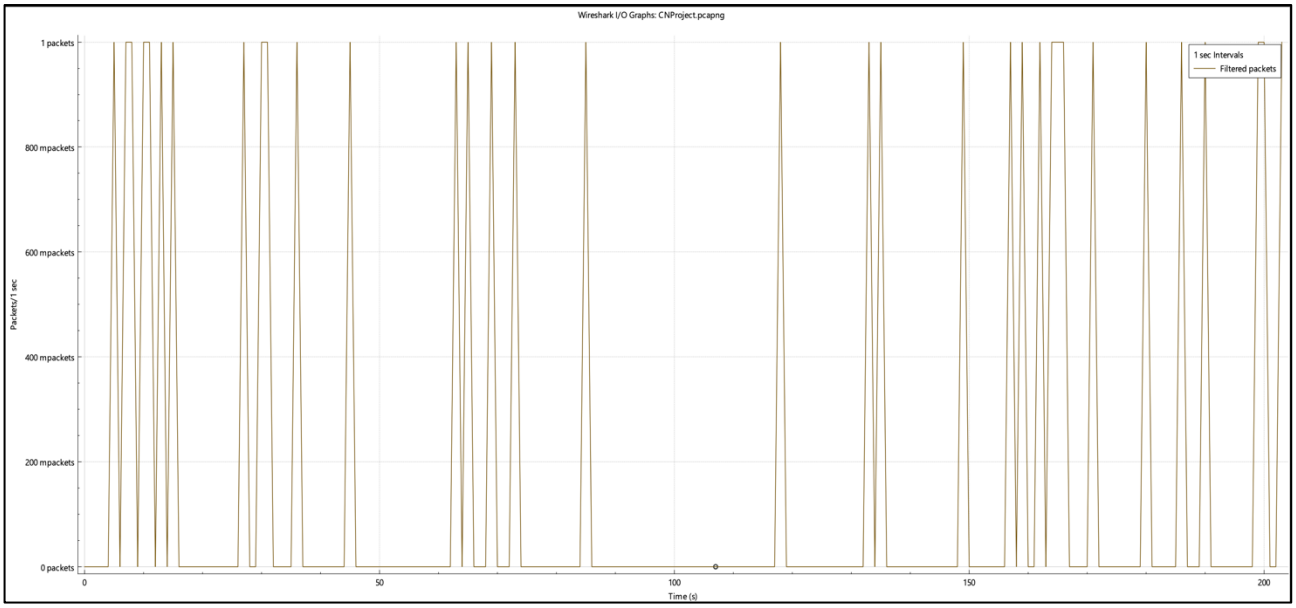


fig. I/O graph demonstrating SNMP packets in the network over time

SNMP Protocol hierarchy:

d) Show the protocol heirachy of the specified protocol & explain all the layers of it.

1] Frame Layer {Layer 1}

- Packets : 34
- Total Bytes : 3053
- Overhead : Includes Frame Check Sequence

2] Ethernet

- Packets : 34
- Bytes : 3053 {Same as frame}
- Header Size : 14 bytes (Ethernet II header)
- Carries high layer protocols.

3] Internet Protocol Version 4 (IPv4)

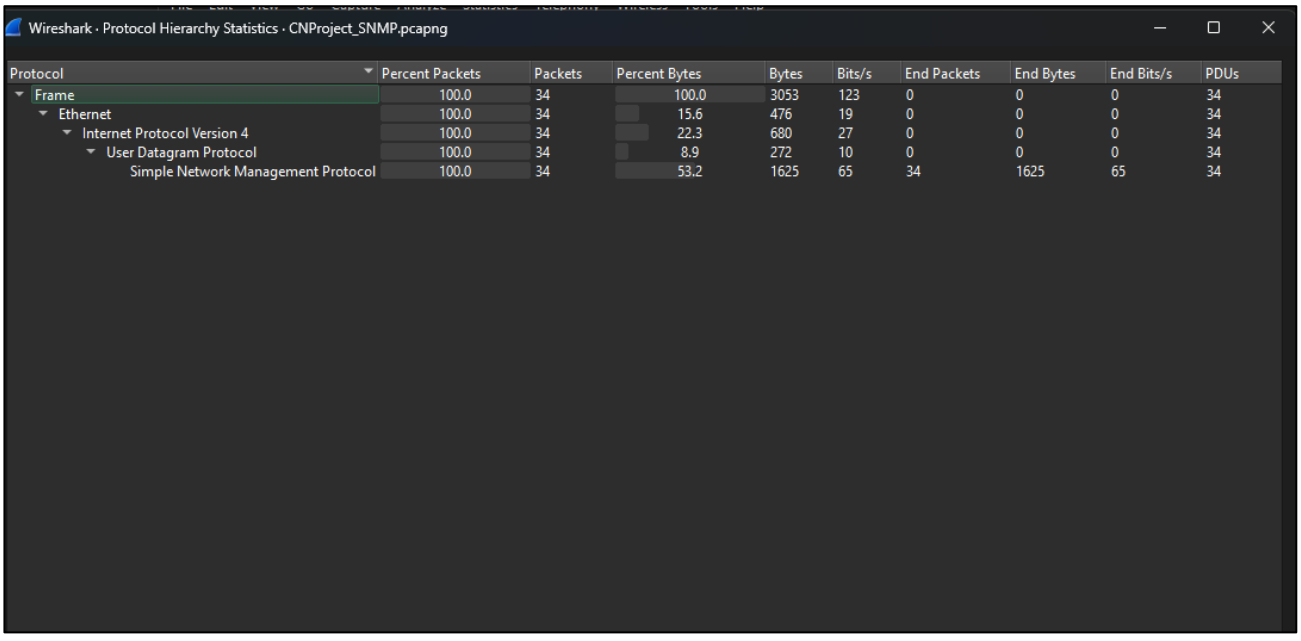
- Packets : 34
- Bytes : 476 (15.6% of total bytes)
- Header Size : Typically 20 bytes {Can be more}
- Encapsulates UDP
- Handles addressing & routing

4) User Datagram Protocol (UDP)

- Packets : 34
- Bytes : 272 (8.9% of total)
- Header Size : 8 bytes.
- Used for it's low overhead

5] Simple ~~layer~~ Network Management Protocol (SNMP)

- ~~Packet~~ Packets : 34
- Bytes : 1625 (53% of total)
- Data Payload : This contains SNMP GET, WALK etc



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	34	100.0	3053	123	0	0	0	34
Ethernet	100.0	34	15.6	476	19	0	0	0	34
Internet Protocol Version 4	100.0	34	22.3	680	27	0	0	0	34
User Datagram Protocol	100.0	34	8.9	272	10	0	0	0	34
Simple Network Management Protocol	100.0	34	53.2	1625	65	34	1625	65	34

Fig. SNMP Hierarchy

Structure of Wireshark:

e) Wireshark is a network protocol analyzer, it allows us to capture & analyse network traffic in real time.

Capture engine : This is responsible for capturing network packets from various interfaces in real time. This operates in the data link layer

Packet dissection engine : This decodes & analyses the captured packets. This engine interprets packet headers & payloads according to various network protocols & displays in human readable format.

USER INTERFACE :

→ Packet list pane: Displays summary of captured packets, showing details like packet number, timestamp, source & destination address, protocol & packet length.

→ Packet details pane: Provides detailed view of the selected packet, showing the contents of each protocol layer in a hierarchical fashion

→ filters: We can apply various filters to display only packets of interest based on specific criteria

→ Analysis tools: There are many built in tools for analysing network traffic, including protocol statistics, flow graphs, etc.

→ Capture file formats: Users can save captured packets in multiple captured file formats

Conclusion

In this project, we successfully simulated and analysed the Simple Network Management Protocol (SNMP) using a MIB browser and Wireshark. The objective was to understand how SNMP operates in real-world communication between managers and agents. Through packet capture and analysis, we explored SNMP's message formats, protocol hierarchy, and traffic patterns.

Wireshark proved to be an effective tool in visualizing SNMP packet flow, investigating protocol layers, and identifying message types. The use of I/O graphs and protocol hierarchies provided valuable insights into SNMP traffic behaviour, confirming its lightweight, request-response nature suited for network monitoring.

This project enhanced our understanding of network protocol analysis and the internal workings of SNMP, laying a solid foundation for further exploration of network management and security protocols.