Remote Network Management Using Python

Thejaswin N, Vikas CV P., Rajeshwari

(Department of Telecommunication Engineering,Dayananda Sagar College of Engineering Bangalore, India, thejaswinvishak@gmail.com)
(Department of Telecommunication Engineering,Dayananda Sagar College of Engineering Bangalore, India, vikascv3062@gmail.com)
(Assistant Professor, Department of Telecommunication Engineering, Dayananda Sagar College of Engineering Bangalore, India, prajeswarisugans@gmail.com)
Navneet Krishna (DGM/D&E/SC&US,Bharat Electronics Limited)

Abstract—

A type of application called remote monitoring and management, sometimes referred to as network management or remote monitoring software, enables managed IT service providers (MSPs) to proactively and remotely monitor client endpoints, networks, and PCs. The phrase "remote IT management" has gained popularity recently. Telnet is a network protocol that allows for remote computer access and the creation of a text-based, two-way communication channel between two computers. It establishes remote sessions using the user-controlled TCP/IP networking protocol (Transmission Control Protocol/Internet Protocol). A network protocol known as Secure Shell, or SSH, creates a secure connection between users and a remote computer.With SNMP, network devices, including routers, servers, and printers, may communicate with network management systems in a standard language (NMS). Through an unsecured network, like the Internet, administrators and other authorised users are able to connect to protected systems.

***Keywords*— NETWORKING, TELNET, SSH, VOIP, SNMP, PYTHON.**

1. INTRODUCTION

One of the features that might help in the problem-finding process is remote network monitoring. With this understanding,remote network administration and troubleshooting may be enhanced. In a word, remote network monitoring makes sure that your networking runs well. Aspects of network administration that are crucial for effective business operations include compliance, reliability, security, and efficiency. To ensure that your IT performs properly, it is crucial to keep an eye out for hardware and software bugs, viruses, and online dangers as well as to stay on top of necessary maintenance tasks. A network protocol called SSH, or Secure Shell creates a secure connection between users and a remote computer.Remote Management is managing a computer or a network from a remote location. It involves installing software and managing all activities on the systems/network, workstations, servers, or endpoints of a client, from a remote location.

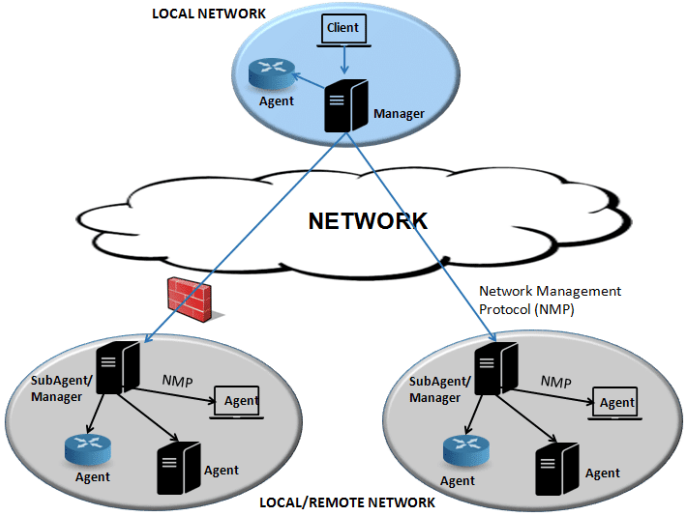

Fig: 1 Local Remote Network Diagram.

This plan intends to create a system where vendors' claims of total supervision and control are as common as the technologies for remote network administration and monitoring. It takes careful investigation and a thorough grasp of business performance and security requirements to identify the tool or tool sets that can offer the all-encompassing coverage your organisation needs.

1.1 Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasises code readability with its use of significant indentation. Its language constructs, as well as its object-oriented approach, aim to help programmers write clear, logical code for small and large-scale projects

1.2 Telnet Protocol

Telnet is an application protocol that enables bidirectional interactive text-oriented communication via the Internet or a local area network utilising a virtual terminal connection. An 8-bit byte-oriented data connection using the Transmission Control Protocol combines user data with Telnet control information (TCP).

1.3 SSH

A network protocol for operating network services securely across an unsafe network is called Secure Shell (SSH). SSH may protect any network service, but common uses include remote command-line, login, and command execution.

1.4 SNMP

An Internet Standard protocol called Simple Network Management Protocol (SNMP) is used to gather, organise, and modify data about managed devices over IP networks in order to alter device behaviour. Cable modems, routers, switches, servers, workstations, printers, and other devices frequently support SNMP.

1.5 VOIP

VoIP, often known as IP telephony, is a technique and collection of technologies for delivering voice conversations and multimedia sessions over IP networks, including the Internet. The provision of communications services (voice, fax, SMS, and voice-messaging) through the Internet as opposed to the public switched telephone network (PSTN), generally known as regular telephone service, is explicitly referred to as Internet telephony, broadband telephony, and broadband phone service (POTS).

1.6 MIB browser

An MIB browser is a tool that allows you to pull data from network devices and displays it in a readable format. It loads MIB files and query data, filters out information from an MIB tree, and gives you the option to configure and manage SNMP traps. Every device vendor provides their own set of MIB files, and an MIB browser can load different MIB files from different vendors. An MIB file contains a description of the object hierarchy on the managed device, as well as the syntax and access privileges for each variable in the MIB.

2. OBJECTIVES

- Reliability- We can expect better stability of hardware and software, and less error rate.
- Managing network resources and services-including the control, monitor, update and device configurations.
- Cost-effectiveness- We aim to reduce the cost by prior planning, modular expansion and relocation of equipment.
- Multiple managers- The monitor can be configured to deal with more than one management station concurrently.

3. BLOCK DIAGRAM

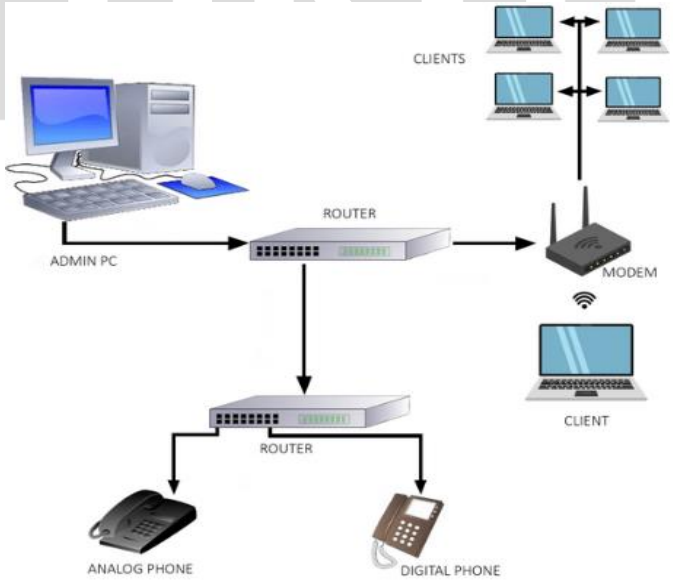

Fig:(3.1) Proposed Block Diagram

4. PUTTY CONFIGURATION

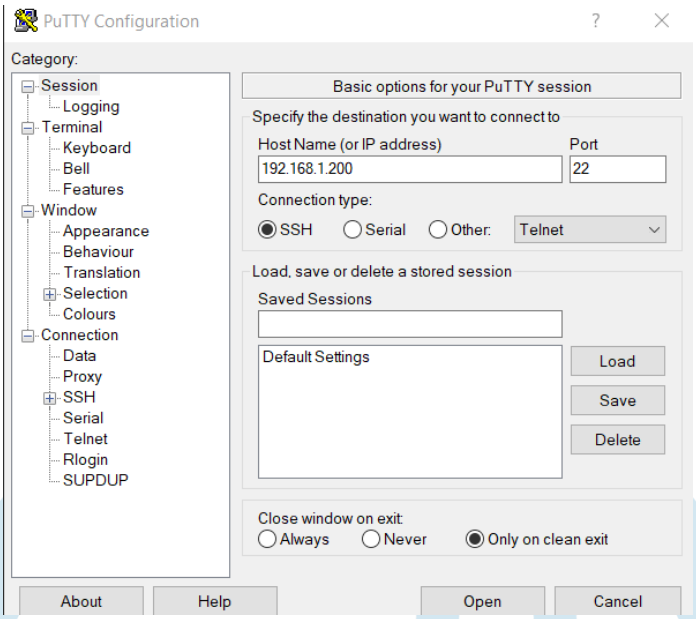

Fig :(4.1) SSH (Putty Configuration)

Step 1**:** Make the setup as per the figure 3.1.

Step 2: Configure the PC ip address to 192.168.1.200/24.

Step 3: Ensure that the lan cable is connected to the switch interface having any ip address.Ex:Gi1/0/1, ip address-192.168.1.100/24.

Step 4: Ensure that ip address 192.168.1.100 should be reachable from the PC by ping command.

Step 5: Open the putty application and select SSH and provide the ip address and port number should be 22.

Step 6: Click on the open and the connection will be successfully established.

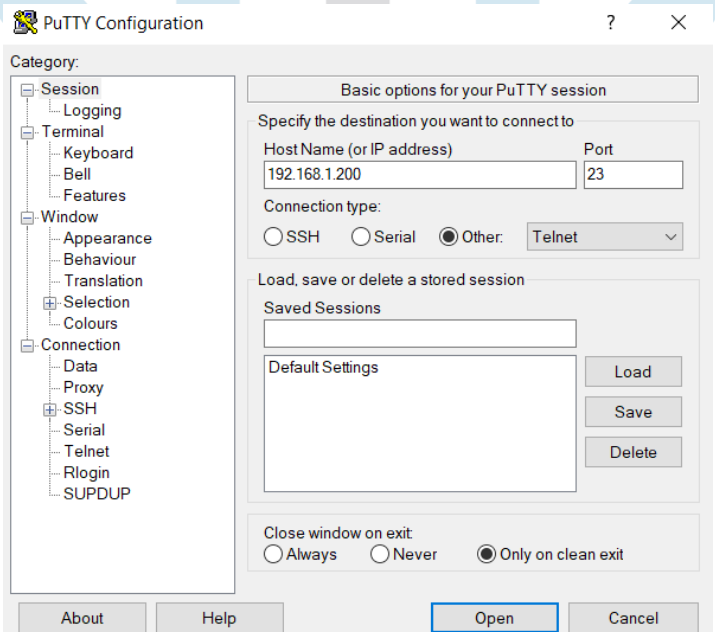

Fig :(4.2) Telnet (Putty Configuration)

Step 1**:** Make the setup as per the figure 3.1.

Step 2: Configure the PC ip address to 192.168.1.200/24.

Step 3: Ensure that the lan cable is connected to the switch interface having any ip address.Ex:Gi1/0/1, ip address-192.168.1.100/24.

Step 4: Ensure that ip address 192.168.1.100 should be reachable from the PC by ping command.

Step 5: Open the putty application and select Telnet and provide the ip address and port number should be 23.

Step 6: Click on the open and the connection will be successfully established.

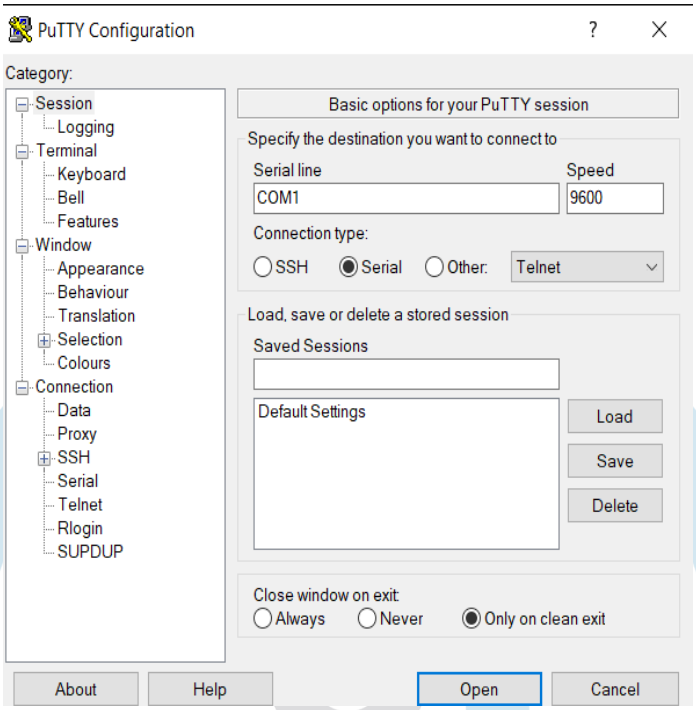

Fig:(4.3) Serial (Putty Configuration)

Step 1**:** Make the setup as per the figure 3.1.

Step 2: Configure the PC ip address to 192.168.1.200/24.

Step 3: Ensure that the Serial USB cable is connected to the switch console port.

Step 4: Ensure that the serial port is detected on the computer with any comport Ex: COM1.

Step 5: Open the putty application and select serial port , COM1 , speed should be given with the respect to device Ex:9600.

Step 6: Click on the open and the connection will be successfully established.

5. FLOW DIAGRAM

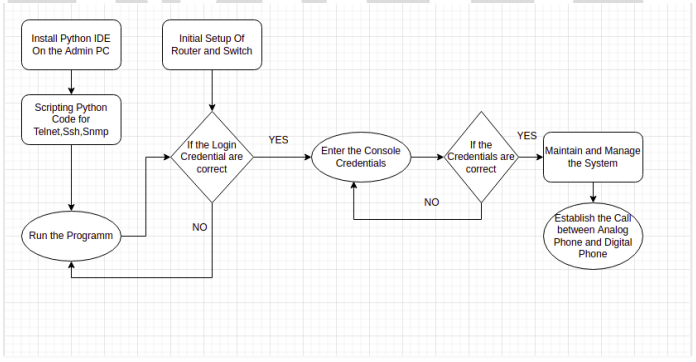

Fig: (5.1) proposed Flow Diagram

It is essential to begin by setting up the Python IDE and doing the fundamental router and switch settings.If the specified Host and IP address are valid, the code connects to the router remotely after it is scripted and run.If the logging credentials are right, access to the console terminal will be granted.We'll need to enter the right password later to gain access to the console terminal.We'll be able to remotely set up, alter, and administer the devices attached to the console terminal once access is granted.Later, we'll construct a call establishment between an analogue and digital phone remotely.

6.SCRIPTINGS

```python
import getpass
import telnetlib

host = "192.168.1.103"
user = input("Enter Username:")
password = getpass.getpass()
print('Successfully passed getpass')
tn = telnetlib.Telnet(host)
print('Successfully passed telnet')
tn.read_until(b"Username:")
tn.write(user.encode("ascii") + b"\n")

if password:
    tn.read_until(b"Password:")
    tn.write(password.encode("ascii")+b"\n")
    tn.write(b"en \n")
    #password = getpass.getpass()
    #tn.write(b"password=Cisco")
    tn.write(b"Cisco\n")
    tn.write(b"conf t\n")
    tn.write(b"int vlan 4\n")
    tn.write(b" no ip add\n")
    tn.write(b" ip add 192.168.1.103 255.255.255.0\n")
    tn.write(b"end\n")
    tn.write(b"exit\n")
    print(tn.read_all().decode("ascii"))
```

Fig:(6.1) Python code for Telnet

```python
import paramiko
import time
import os
import sys

hostname = "192.168.1.102"
router_username = "cisco"
router_password = "Cisco"
port="22"
ssh = paramiko.SSHClient()
print("Successfully Host")

# Add SSH host key automatically if needed.
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())

ssh.connect(hostname=hostname,username=router_username,password=router_password, look_for_keys=False, allow_agent=False)
print(f"Successfully connected ")
#
remote_conn = ssh.invoke_shell()
print("Interactive SSH session established")

output = remote_conn.recv(1000).decode()
print(output)

remote_conn.send("en\n")

output = remote_conn.send("Cisco\n")
time.sleep(5)

output = remote_conn.recv(5000).decode()
print(output)

remote_conn.send("\n\n")
remote_conn.send("conf t\n")
time.sleep(2)
output2 = remote_conn.recv(5000).decode()
print(output2)

remote_conn.send("do show run\n")
time.sleep(5)
output2 = remote_conn.recv(5000).decode()
print(output2)
ssh.close()
print("Successfully closed")
```

Fig:(6.2) Python code for SSH

```python
import os, sys
import socket
import random
from struct import pack, unpack
from datetime import datetime as dt

from pysnmp.entity.rfc3413.oneliner import cmdgen
from pysnmp.proto.rfc1902 import Integer, IpAddress, OctetString
#from pysnmp.proto.rfc1213 import Integer, IpAddress, OctetString
ip='192.168.1.102'
community='pub'
value=(1,3,6,1,2,1,1,1,0)

generator = cmdgen.CommandGenerator()
comm_data = cmdgen.CommunityData('server', community, 1) # 1 means version SNMP v2c
transport = cmdgen.UdpTransportTarget((ip, 161))

real_fun = getattr(generator, 'getCmd')
#res = real_fun.getCmd(oid='1.3.6.1.2.1.1.1.0', Version=2, DestHost='192.168.1.102', Community='pub')
res = (errorIndication, errorStatus, errorIndex, varBinds)\
    = real_fun(comm_data, transport, value='1.3.6.1.2.1.1.1.0')
print("Successfully")
print(res)
```

Fig:(6.3) Python code for SNMP(Get)

```
Router:
    enable
    configure terminal
    hostname Net_rtr
    interface fastethernet0/0
    ip address 192.168.100.1 255.255.255.0
    no shutdown
    exit
        ip dhcp pool Voice
        network 192.168.100.0 255.255.255.0
        default-router 192.168.100.1
        option 150 192.168.100.1
        exit
            ip dhcp excluded-address 192.168.100.1
            telephony-service
            max-dn 4
            max-ephone 5
            ip source-address 192.168.100.1 port 2000
            auto asign 1 to 4
                ephone-dn 1
                number 1234
                ephone-dn 2
                number 5678

Switch:
    enable
    configure terminal
    hostname Net_Sw
    interfacerange fastethernet0/1-24
    switchport mode access
    switchport voice vlan 1
    end
    copy run start
```

Fig:(6.4) Cisco Code for Call Establishment

7. RESULT

Fig :(7.1) Output of Telnet

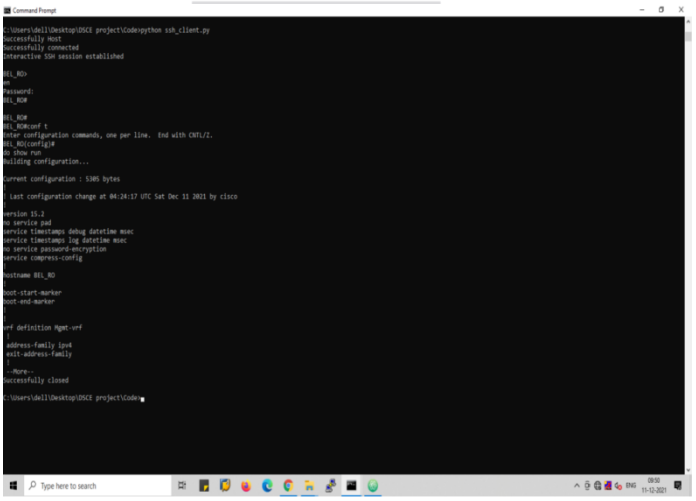

Fig :(7.2) Output of SSH

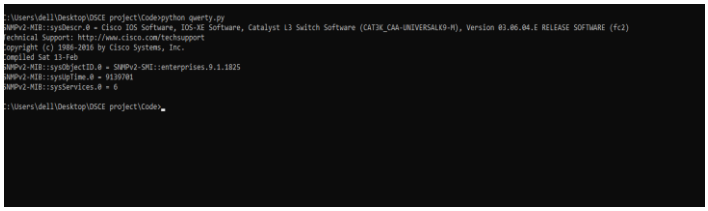

Fig :(7.3) CMD output of SNMP(Get)

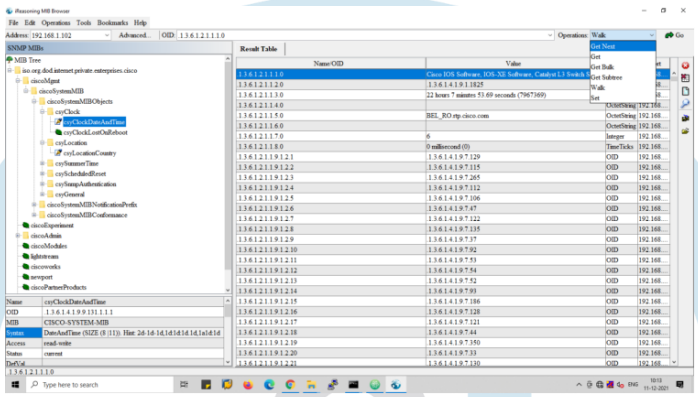

Fig :(7.4) Output of SNMP using Mib Browser

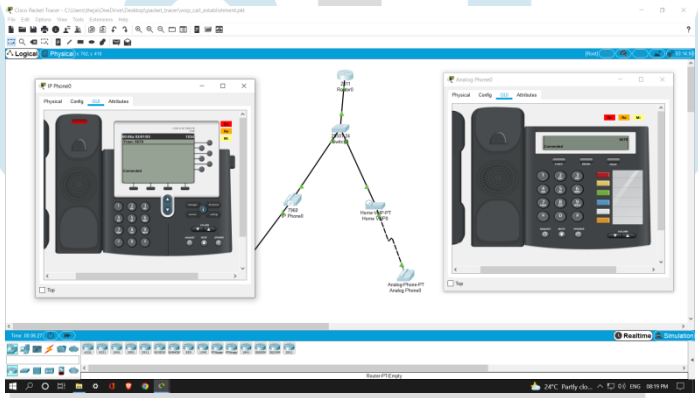

Fig :(7.5) Simulated (Cisco Packet Tracer) output of Call Establishment

8. CONCLUSION

In conclusion, the project objectives were met and the remote network management system was effectively constructed. The firm should put the suggested design for a network management system into practice. The requirement for a network management system and its particular generic implementation has been covered. Due to its straightforward architecture and implementation, telent is quickly replacing ssh as the industry standard for remote administration and logins. The suggested paradigm would provide effective network administration and a seamless update of network services. The block diagram has also been briefly suggested as a viable approach for next-generation network administration. As a result, we have advised that these upgrades be made as soon as possible to improve the network's security, functionality, and dependability.

References:

1] W. Zeng and Y. Wang, "Design and Implementation of Server Monitoring System Based on SNMP," 2009 International Joint Conference on Artificial Intelligence, 2009, pp. 680-682, doi: 10.1109/JCAI.2009.34.
[2] B. Goode, "Voice over Internet protocol (VoIP)," in Proceedings of the IEEE, vol. 90, no. 9, pp. 1495-1517, Sept. 2002, doi: 10.1109/JPROC.2002.802005.

[3] K. W. Hong, W. Ryu and J. K. Choi, "Telnet-based transport control," 2009 11th International Conference on Advanced Communication Technology, 2009, pp. 887-889.
[4] Paul Mihăilă, "Network Automation and Abstraction using Python Programming Methods", https://www.researchgate.net/publication/322017645_Network_Automation_and_Abstraction_using_Python_Programming_Methods.
[5] T. Ylonen, "SSH Key Management Challenges and Requirements," 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019, pp. 1-5, doi: 10.1109/NTMS.2019.8763773.
[6] P. Čeleda, P. Velan, B. Král and O. Kozák, "Enabling SSH Protocol Visibility in Flow Monitoring," 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2019, pp. 569-574.
[7] He Peng, Qiu Jianlin, Gu Xiang. Design and Implementation of Remote Host Monitoring System Based on SNMP. Computer Engineering and Design. 2008, 29(13):3303-3312.

IJRTI