TRIBHUVAN UNIVERSITY

Prime College

Nayabazar, Kathmandu, Nepal

A Project Report On

"NetSec Application" (Network Security Application)

Submitted By

Abishek Khanal (15303)

Dibya Darshan Khanal (15314)

Nirajan Shahi (15322)

Yogesh Prajapati (15349)

A Project Report Submitted in partial fulfillment of the requirement of **Bachelor of Science in Computer Science & Information Technology (BSc.CSIT) 5th Semester** of Tribhuvan University, Nepal

February
2020

NetSec Application

A project report submitted for the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer science & Information Technology awarded by Tribhuvan University.

Submitted By

Abishek Khanal (15303)

Dibya Darshan khanal (15314)

Nirajan Shahi (15322)

Yogesh Prajapati (15349)

Submitted To

Prime College

Department of Computer science

Affiliated to Tribhuvan University

Khusibun, Nayabazar, Kathmandu

ACKNOWLEDGEMENT

We want to express our most profound appreciation to all those who provided us with the possibility to complete this report—special gratitude we give to 5th-semester project Supervisor, **Mr. Hiranya Pd. Bastakoti**, in whose contribution stimulating suggestions and encouragement, helped us to coordinate our project, especially in writing this report. Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of Prime College, who permitted to use all required equipment and the necessary materials to complete the task. We are thankful and fortunate enough to get constant support from our seniors and every teaching staff of B.Sc. CSIT department which helped us achieve our project. We would also like to extend our regards to all the non-teaching faculty of B.Sc. CSIT department for their timely support. We have to appreciate the guidance given by another supervisor as well as the panels, especially in our project presentation that has improved our presentation skills thanks to their comment and advice. Our thanks and appreciations also go to each one of our colleagues for their encouragement and support in developing the project.

With respect,

Abishek Khanal (15303)

Dibya Darshan khanal (15314)

Nirajan Shahi (15322)

Yogesh Prajapati (15349)

ABSTRACT

NetSec Application is a network application comprising security features. This very application will connect to a network device; in this case, we have Arista Switch via SSH. In its broadest sense, NetSec Application aims the practices and technology a business puts in place to protect its IT infrastructure. This infrastructure, in turn, is made up of all the data, programs, applications, web networks, software and hardware utilised and managed by the business.

Network security is one of the most important aspects to consider when working over the internet, LAN or another method, no matter how small or big your business is. While no network is immune to attacks, a stable and efficient network security system is essential to protecting client data. A sound network security system helps business reduce the risk of falling victim of data theft and sabotage.

People use SSH to communicate securely with another computer. By using SSH, the exchange of data is encrypted across the Internet pathways. This way, anyone who happened to see the data streaming by, would not be able to see what was in the data.

KEYWORDS: SSH, Network, Security, Communication

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

SSH- Secure Shell

NetSec- Network Security

IETF - Internet Engineering Task Force

IP - Internet Protocol

R-LOGIN– Remote Login

RSH- Remote Shell

DNS- Domain Name System

GUI – Graphical User Interface

OS – Operating System

CMD – Command

CLI – Command Line Interface

DFD - Data Flow Diagram

UML - Unified Modeling Language

CHAPTER 1. INTRODUCTION

Background

Security is being a significant threat to everyone connected to the network. In this very era; network security is must to consider. Network security is any activity designed to protect the usability and integrity of your system and data. Security includes both hardware and software technologies. It targets a variety of threats. NetSec Application stops threats them from entering or spreading on your network. It is sufficient to network security to manage access to the network.

One of the easiest ways to secure a network is to make sure you have some simple security features enabled on switch ports to which your end users connect. Properly securing switch access ports is reasonably straightforward, but some techniques are often overlooked. Secure Shell (SSH) is a network protocol designed initially by Tatu Ylönen, a researcher at Helsinki University of Technology, Finland, as a replacement for earlier protocols such as TELNET, FTP and rlogin. SSH version 2(SSH-2) is a revised version of the protocol by the Internet Engineering Task Force (IETF). SSH is primarily favored over other protocols since it uses public-key cryptography to provide a secured connection. SSH uses the client-server model. SSH primarily uses port no 22.
For this very project we have built an SSH tool that secures intruders running in the switch unauthorized, it will capture data from switch and provide them to remote servers. The dynamic graph show-down will give clear view of data being captured in real time.

Problem Definition

Routers, switches and firewalls, collectively known as Network Infrastructure Devices, are the most essential elements of any network. In terms of security, they *should* be the most hardened devices; however, due to their importance; administrators are seldom inclined to update them to ensure they do not inadvertently affect any network uptime.
NetSec Application is a broad tool, and it's full in the sense that your network administrator can implement it anywhere necessary; it's not confined to only one purpose. Secure Shell implemented in our application is a program to log into another computer over a network, to execute commands in a remote machine, and to move files from one device to another. It provides strong authentication and secure communications over insecure channels. It is a replacement for R-LOGIN and RSH.
SSH protects a network from attacks such as IP spoofing, IP source routing, and DNS spoofing. An attacker who has managed to take over a network can only force SSH to disconnect. He or she cannot playback the traffic or hijack the connection when encryption is enabled.
When using SSH's login (instead of rlogin) the entire login session, including transmission of password, is encrypted; therefore it is almost impossible for an outsider to collect passwords.

Objectives

This project aims to meet the following objectives:

- To create an SSH connection which is used for managing routers, server hardware, virtualisation platforms, OS, and inside systems management and file transfer applications.
- To validate and verify IP addresses and IP tables.
- To connect to servers, make changes, perform uploads and exit, either using tools or directly through the terminal.
- Secure management of network infrastructure components like Switches, Routers and Vlans.

Scope

Very often we need to work on a virtual machine when developing large projects or another practical example, writing a plan in which we are highly dependent on the hardware and the operating environment, and we write the software on another computer. A very convenient solution to this case is the use of remote access in the development environment, in the NetSec Application case; we can use SSH connection in our virtual Arista Switch.
IP validity checks overall IP reachability along with the switch and allows those credentials to log users in the system.
A bright and contrast view of the graph in the Application shows the data in and out from the virtual switches, which make Network admin analyse data packets.
However, the credentials set on the switches should be out of reach of intruders, they should be protected or changed frequently.

Limitations

The limitations of this project can be listed out as follows:

- The SSH shell has no GUI based interface; it might be hard to understand to non-technical member.
- If you run the NetSec application, you will need an SSH-client. Linux already has spawn and remote client for SSH.
- Our application is only confined to connect to one remote client; in our case, we have windows as the only client.
- Extending NetSec Application reachability will use more resources on the client-server.

CHAPTER 2. REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

Literature Review

Authentication plays an essential role in the security of innovative Internet-based applications. Authentication is critical in two-party and multi-party communications and is of several kinds: user authentication, remote user authentication, mutual authentication, message authentication, and implicit authentication. A significant part of the proposed work is about Authentication, and we present a subset of the latest research relating to this chapter. Current research on authentication is presented along the lines of hash-based, digital signature-based, id-based, hierarchical model-based, Subscriber Identity Module (SIM) based, three party-based, groups-based supporting greater than three parties, and mutual authentication.

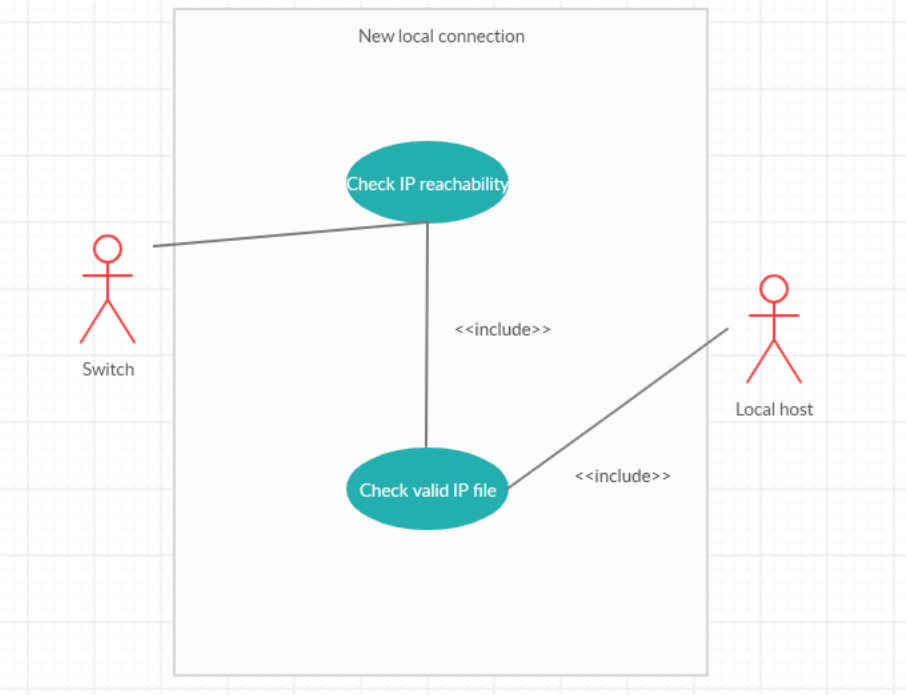
Requirement Specification

Functional Requirement

Functional requirements define the statements of services that the system should provide first hand and how the system should react to behave to the particular inputs and particular situations. The functional requirements of NetSec application are defined with the use of Use-case diagrams as follows:

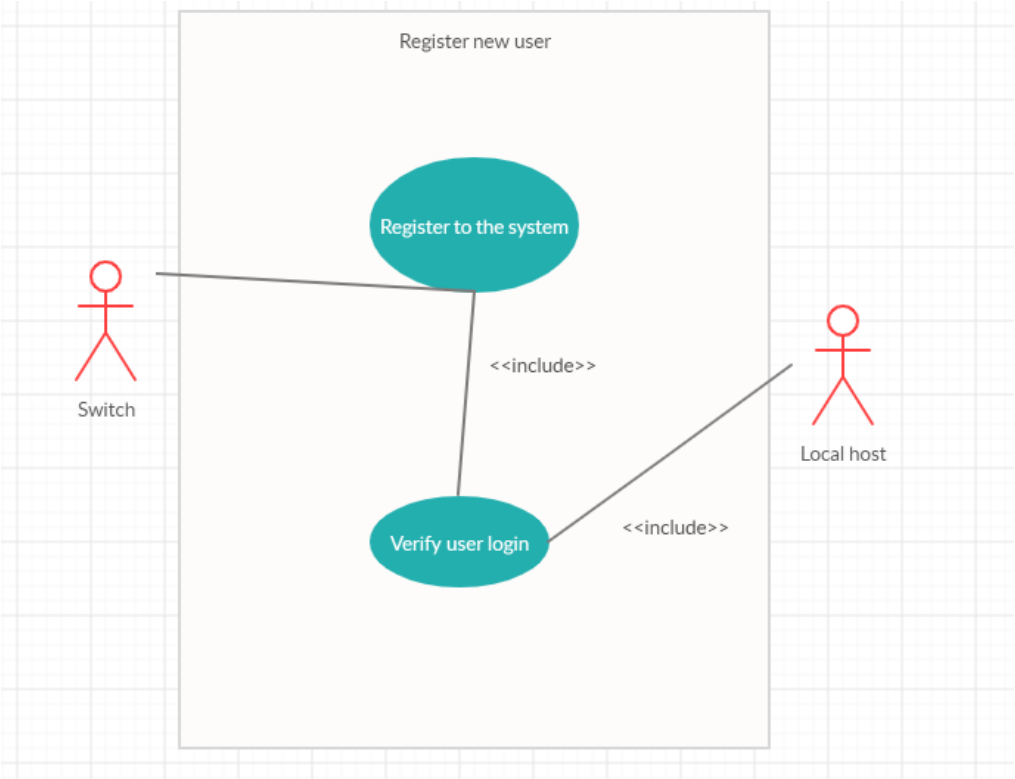
Check IP address validity

The system checks for the valid IP address from the IP table and file included in the project.



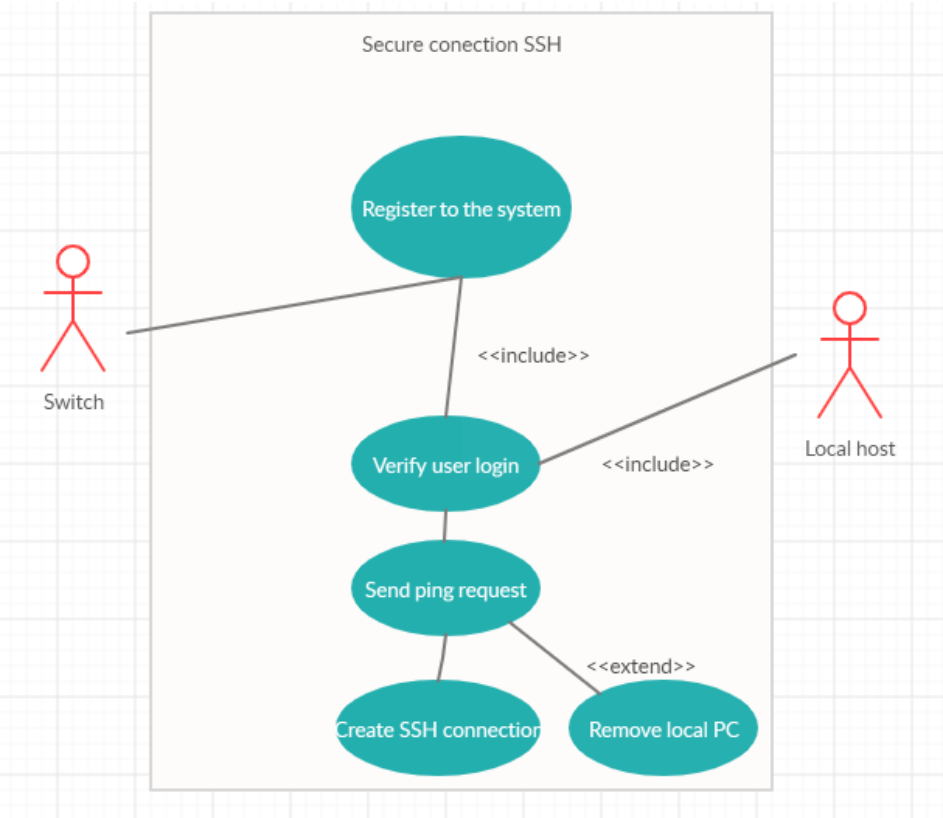
Register User

The system shall authenticate the user after registering to the system with the required email address and desired password.



Create SSH connection

This requirement provides the Arista switch by providing a secure connection with the local machine.



Non-Functional Requirement

- **Concurrency**: Since NetSec Application is a real-time application, it has the potential to handle multiple ping request and should execute simultaneously for each of the Localhost machines.

- **Reliability:** All the data, including the user data and the traffic data, are stored in the secure shell which system admin can analyse.

- **Usability:** With the provided connection, the application is simple and convenient to use where admin can conveniently use.

Feasibility study

The feasibility study is the initial design stage of the project that indicates if the proposed project is possible or not. It determines if the proposed project is useful or not and defines the practicality of the project or a system. The study also identifies if the system can be built efficiently on the required time with the available resources.

Technical feasibility

The technical feasibility study focuses on gaining an understanding of the existing technological resources and their applicability to the expected needs of the proposed system. All the tools and software product that is required for this project is readily available on the internet. Required tools and iso images for Switches are easily available. However, the implementation of the algorithm and the calculations are rather complex.

- **Hardware Requirements:**

The application is for the PC, and hence PC which uses Linux or Windows with minimum requirement to run Virtual Box is capable of running all the files.

- **Software Requirements:**

The application is running on Python so required python libraries and installation is required. Virtual boxes with exact iso images of Arista switches are used.

Operational feasibility

Operational feasibility measures how well a proposed system can solve the defined problem, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase. The system can be developed to be reliable, maintainable, usable, sustainable and affordable. So, this system is operationally feasible.

Economic feasibility

Economic feasibility analyses the project's costs and revenue to determine whether it is possible to complete or not. Although no extra equipment was bought for conducting our project, we had to select a suitable plan for configuring and using Virtual box provided us with a level of configuration to the system admin. The potential cost that may be added upon the system, could be that of as the proposed project requires no extra equipment to be bought and the required needs are already present, so the system is economically feasible.

Schedule Feasibility

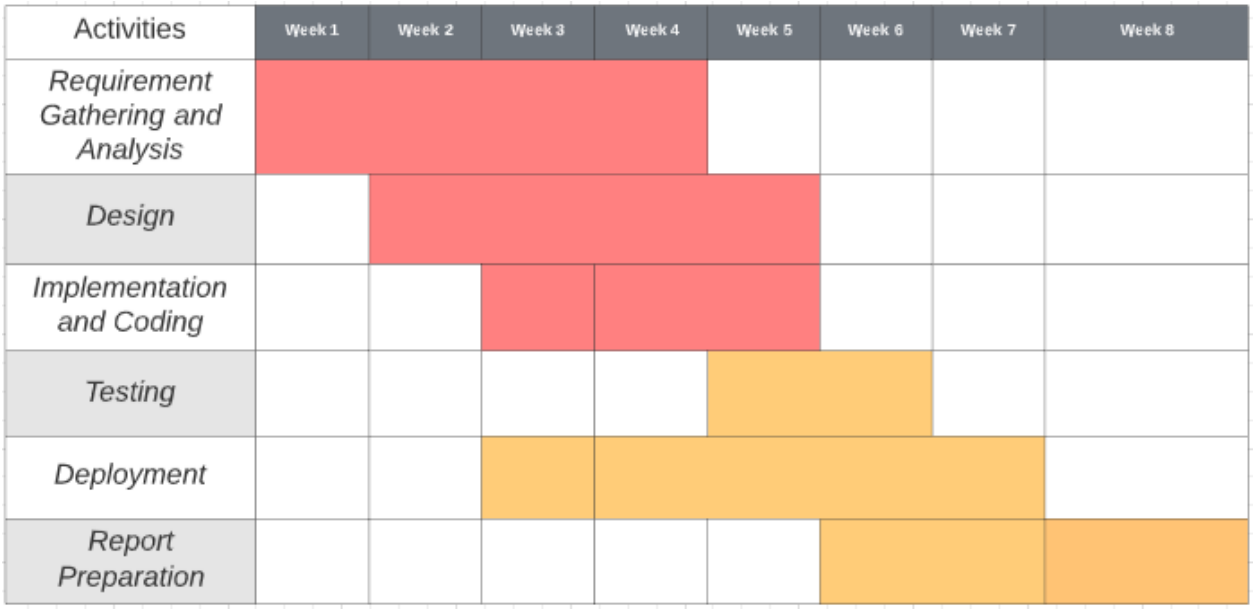


Table: Gantt chart

The above Gantt chart displays the overall timeline of the project. It presents the sequential breakdown of the task involved in the project with the time taken for each job. The first two activities were carried out per schedule. Implementation and coding phase was carried out in parallel with the report preparation. The testing phase of the project was carried along with the documentation until the completion of the project.

Structuring System Requirements

Process Modeling (DFD Level-0 and Level-1)

Context Level Diagram (DFD level-0)

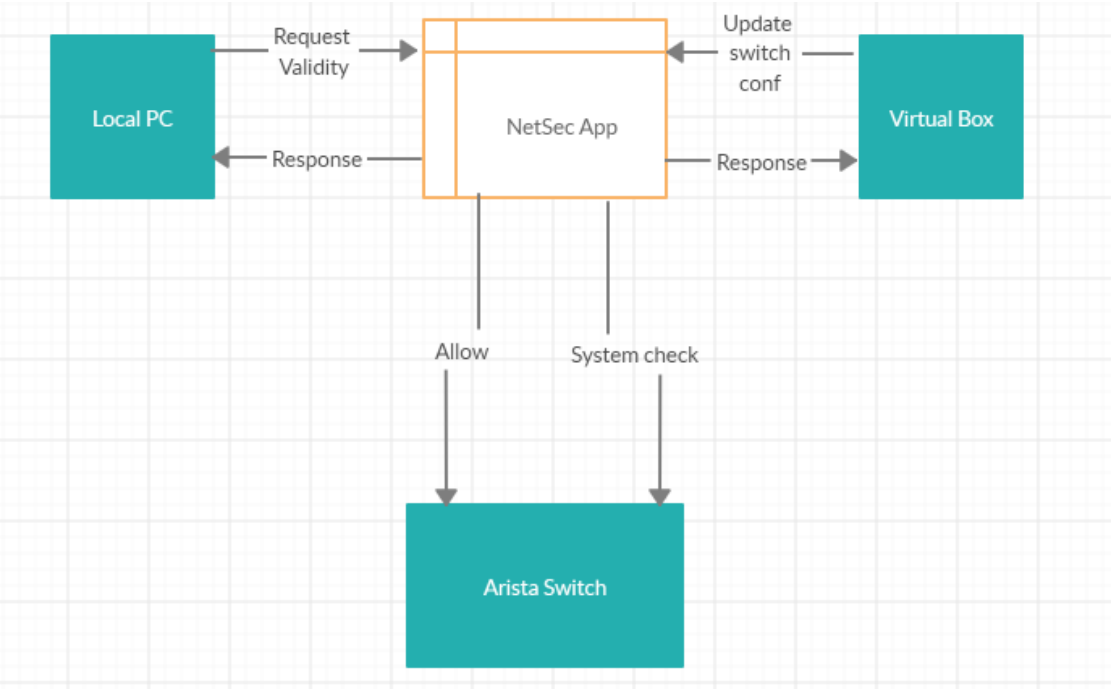


Figure: Context level diagram

The above context level diagram shows the overall data flow throughout the system where commuter and the contributor work in parallel. The commuter request for the best path. The contributor updates the traffic conditions to the system and the admin verifies all the data running through the system.

Context Level Diagram DFD Level-1

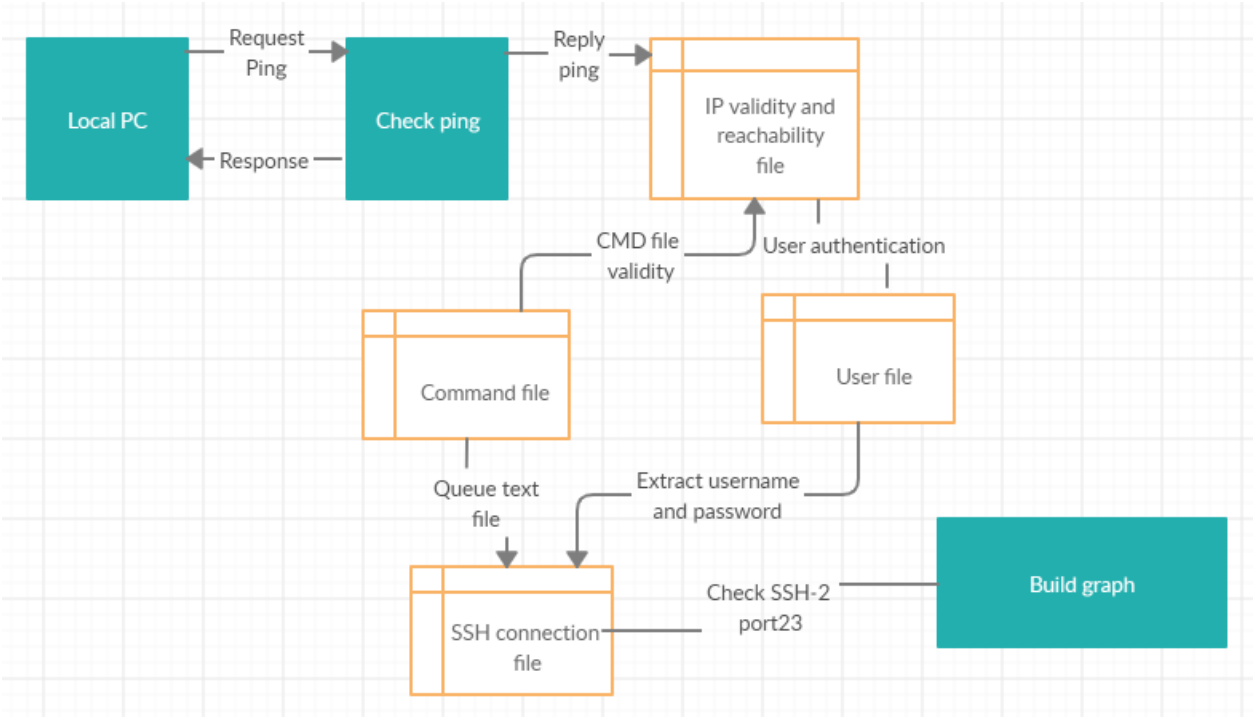


Figure: DFD level-1 diagram

The DFD Level-1 diagram shows the inner process of the system which includes navigation, traffic data controller and authentication system. The diagram depicts the flow of data throughout the system.

CHAPTER 3: SYSTEM DESIGN

We have developed the required system that works with the help of the Internet. To use this system, we need a user with an understanding of computer systems and the Internet. The above architecture shows how different users in the system exchange messages through the interaction of Web service agents in each system.

System Design

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

UML (Unified Modeling Language) class diagram:

In software engineering, a class diagram in the UML is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

Logical flow diagram

Overview

We have developed the required system that works with the help of the internet. to run the system we first check IP address is valid or not, if IP is correct we check the IP is reachable during ping, then we establish a connection via SSH connection, in this session we can capture packet from a reachable IP address that has been already stored in virtual switch, by this way we can capture packet while ping to the reachable Ip and also can plot the graph depending upon the received packet.

Algorithm used

We have different algorithm and scripts in different part of the system to develop the functionalities present.

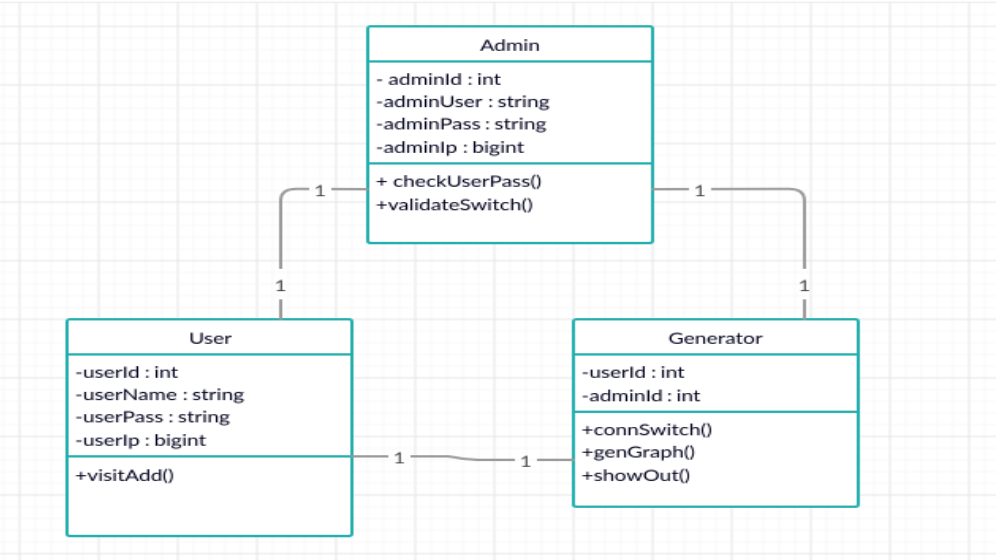
To check if IP address is valid or not

```
import sys
#Checking octets
def ip_addr_valid(list):

   for ip in list:
      ip = ip.rstrip("\n")
      octet_list = ip.split('.')

      if (len(octet_list) == 4) and (1 <= int(octet_list[0]) <= 223) and (int(octet_list[0]) != 127) and (int(octet_list[0]) != 169 or int(octet_list[1]) != 254) and (0 <= int(octet_list[1]) <= 255 and 0 <= int(octet_list[2]) <= 255 and 0 <= int(octet_list[3]) <= 255):
         continue

      else:
         print('\n* There was an invalid IP address in the file: {} :(\n'.format(ip))
         sys.exit()
```

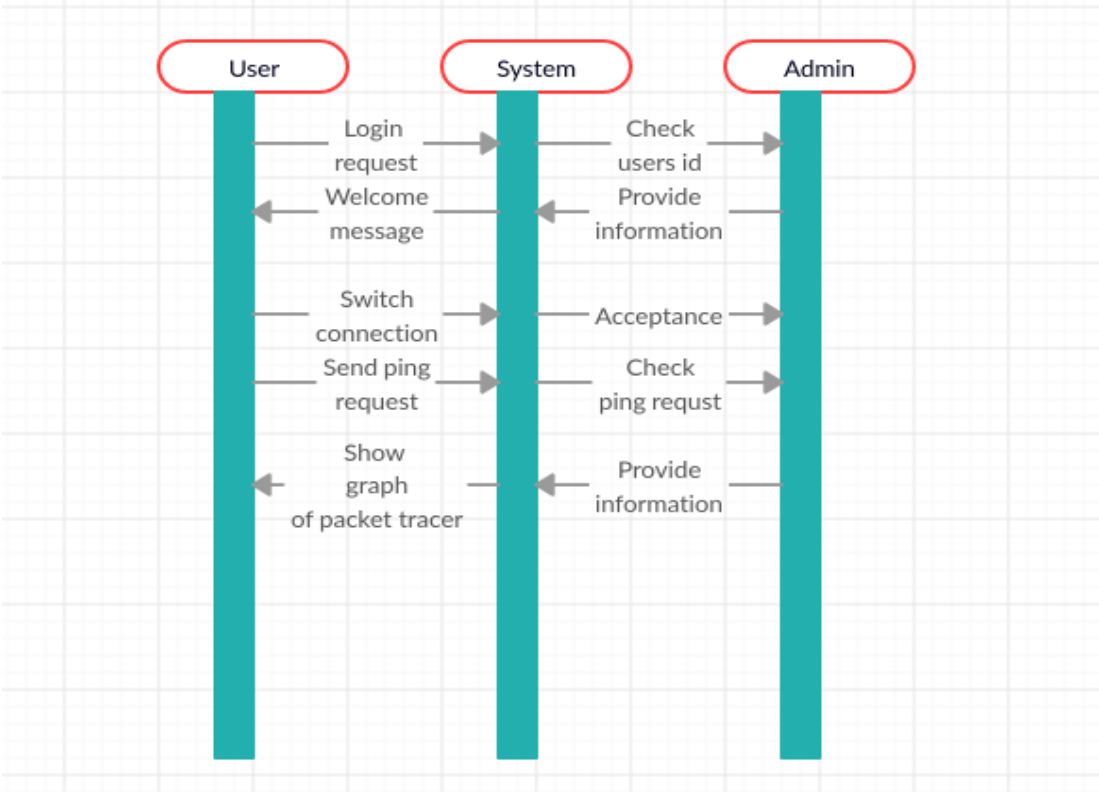
To check if IP is reachable or not

```
import sys
import subprocess

#Checking IP reachability
def ip_reach(list):

   for ip in list:
      ip = ip.rstrip("\n")

      ping_reply = subprocess.call('ping %s -n 2' % (ip,), stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)

      if ping_reply == 0:
         print("\n* {} is reachable :)\n".format(ip))
         continue

      else:
         print('\n* {} not reachable :( Check connectivity and try again.'.format(ip))
         sys.exit()
```

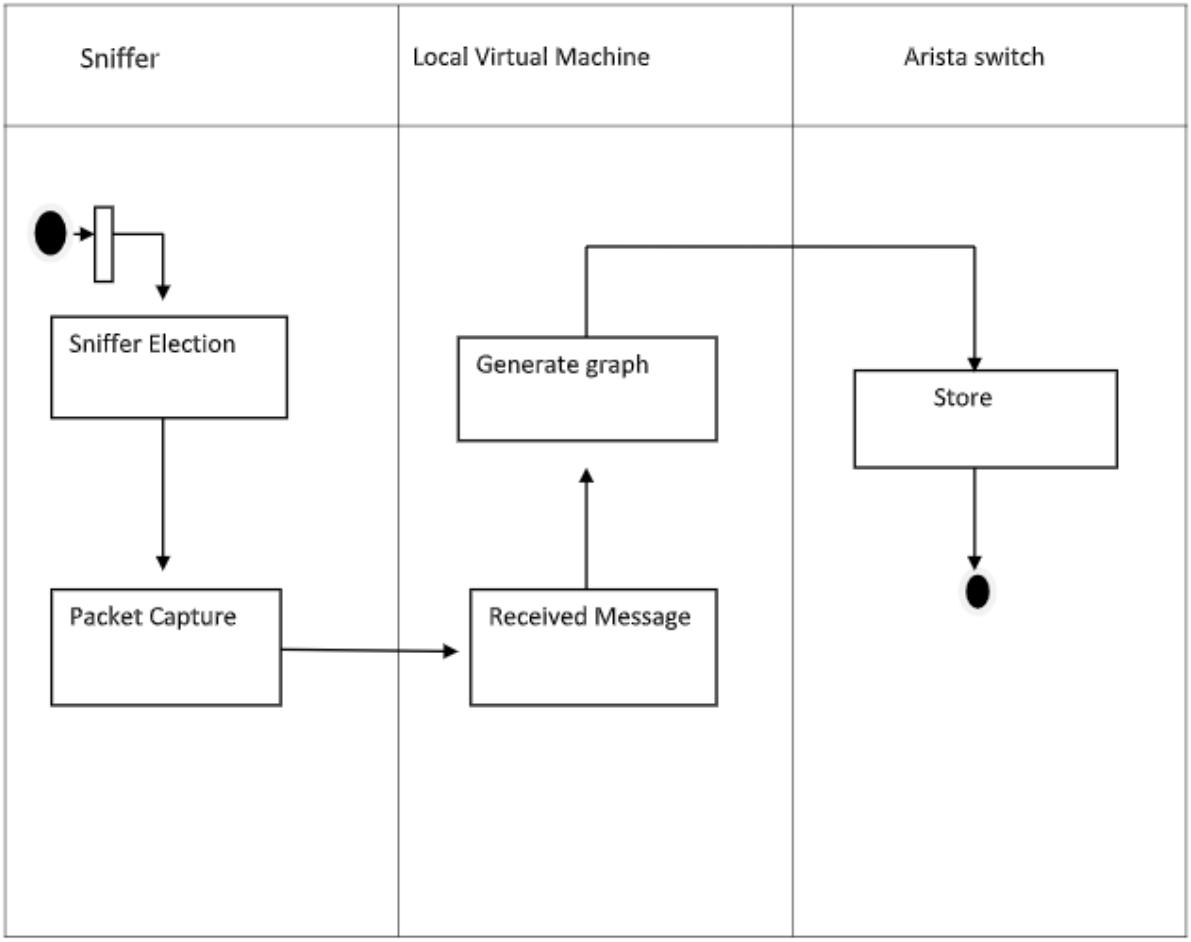
Class diagram



Sequence Diagram



Activity Diagram



CHAPTER 4. SYSTEM IMPLEMENTATION AND TESTING

Implementation Overview

The Incremental Development model is an incremental approach for developing a software product as the product is designed, implemented and tested incrementally until the product is finished. This approach for developing software applies to the waterfall model progressively. The incremental development model is used where the requirements are clear before the development of the product and can be implemented in phases.

Since this project doesn't deal with any clients, incremental development is used so, all the requirements are clear. Development of the software is carried by tackling the highest priority requirement and goes through each development phases for each of the requirements until the final product is delivered.

We are using incremental development model as each increments for the specific requirements can be reviewed for the required changes or modification.

Tools Used

Front End Tools

Local PC: We are using Local PC as a client that receives switches data through authorized login and setup. Local PC for testing gives us flexibility to run codes and commands with ease. We are using Windows OS for this purpose. We are developing our backend python codes in Notepad++ then running it through Windows CMD interface. Local PC's IP is set to 10.10.10.1; it's also a gateway to our network connection.

Virtual Box: Virtual Box is great tool when it comes for testing any applications. We are using Arista Switch which is freely available in their website for test purpose. Oracle Virtual Box is also an open source tool which is great deal for us. We have setup two Arist switches named, Arista 1 and Arista 2.

Arista 1: Arista 1 switch is set with bridged connection to our TP-link Wi-Fi adapter. It's IP is set to 10.10.10.2 followed by Subnet Mask 255.255.255.0 through CLI.

Arista 2: Arista 2 switch is also set with bridged connection to our TP-link Wi-Fi adapter. It's IP is set to 10.10.10.3 followed by Subnet Mask 255.255.255.0 through CLI. This switch is essentially used to send heavy packets to Arista 1 through CLI: *ping 10.10.10.2 size 18000 repeat 25000*

Back End Tool

Python- Python is a great tool to build network and security application. Python is rich in libraries for building applications. We can interact with switches and route through secure SSH connection with Netmiko and Paramiko library function. Python is flexible to run while setting the required environment and editors.

CLI- Command Line Interface is a terminal-based interface which interacts with the hardware itself. In order to maintain socket programming for the application, we can easily set IP and subnet to each switch. Command-line interface is a real backend tool to use for internal programming and setup.

Modules Description

This project can be decomposed into the following modules:

a) **IP file validity**: This module is responsible for checking valid IP addresses. It must be valid according to the Classes and Reserved IP address.

b) **IP address validity**: This module is responsible for checking the IP address in the IP file table to allow authorize IP to access to the switch.

c) **IP_reach module**: This module is responsible for checking IP reachability according to subprocess and system.

d) **SSH connection**: This module contains all the previously mentioned methods. This module is responsible to connect remote server and Arista Switch to transfer packets. All the above methods build a secure connection through check and balance with IP.

e) **Graph:** It is a graphical representation of the data being captured in the remote server (Windows OS). It is created using Matplotlib python library,

The **python libraries** used in this project are as follows:

- sys
- time
- threading
- paramiko
- datetime
- os.path
- subprocess
- time
- sys
- re
- matplotlib.animation
- matplotlib

Methods used:

- ip_reach(list):
- ip_file_valid():

- ip_addr_valid(list):
- animation_function(i):
- subplot.clear()
- subplot.plot(x)
- pyp.show()
- ssh_connection(ip):
- session.close()
- session = paramiko.SSHClient()
- selected_cmd_file.seek(0)
- create_threads(list, function):

Testing

Software testing is the process or an activity to check whether the actual results match the expected results and to ensure that the software system is free of errors. Testing is carried out during the development of the software.

Unit Testing

Unit testing is the part of the testing methodology which includes testing of individual software modules and the components that make up the entire software. Each module or the components are individually tested to check for any kind of errors or misbehaviors.

Table 1: Test case for Username/password

SN	Test Case ID	Test Description	Input Test Data	Expected Result	Actual Result	Remarks
1	TC-INS-01	Test for user file path and name	Set path to D:\NetSec\ user.txt	Username/password file is valid	Application executed with Username/password file is valid.	Pass

Table 2: Test case for Command

SN	Test Case ID	Test Description	Input Test Data	Expected Result	Actual Result	Remarks
1	TC-LG-01	Test for CMD file path and name	Set path to D:\NetSec\cmd.txt	Command file is valid	Application executed with command file is valid.	Pass

Table 3: Test case for IP

SN	Test Case ID	Test Description	Input Test Data	Expected Result	Actual Result	Remarks
1	TC-LG-01	Test for IP file path	Set path to D:\NetSec\ip.txt	IP file is valid	Application executed with IP file is valid.	Pass

Table 4:Test case for IP reachability

SN	Test Case ID	Test Description	Input Test Data	Expected Result	Actual Result	Remarks
1	TC-LG-01	Test for IP reachability	No input required	10.10.10.2 is reachable	Application executed with Done for device 10.10.10.2l. Data sent at (Date) (Time).	Pass

System Testing

System testing is the part of the testing methodology that includes the testing of the entire integrated system for any kind of errors and bugs. System testing is basically done to check if the system works when executed. This test is carried out by interfacing the components of the entire system and testing it as a whole.

Table 6: System Testing

SN	Test Case ID	Test Description	Input Test data	Expected result	Actual Result	Remarks
1	TC-INS-01	Send heavy packets to Arista 1 from Arista 2	Ping 10.10.10.2 size 18000 repeat 25000	Sending (some) bytes from 10.10.10.3 icmp (data) ttl= (data) time= (time)	Application executed Sending (some) bytes from 10.10.10.3 icmp (data) ttl= (data) time= (time)	Pass
2	TC-INS-02	Test for graph	Python D:\NetSec\ graph.py	Graph plotted, named figure 1	Graph successfully executed (graph point increases according to number of packets sent)	Pass

CHAPTER 5. CONCLUSION

Significance

After the implementation and testing of application with network; it became easy as there were no complex routing interaction. It was seen to it that servers could reach out to any point of the network i.e. connectivity within all servers was ensured. More so the application could be made multi-platform support without major design changes i.e. it can be flexible at any instant.

Conclusion

The development of this application provides a platform for users to help fellow network administrator to create secure tunneling with the client. We expect the application to work properly on the installed systems i.e. windows and Linux. Provided the necessary conditions, like a stable network connection or internet connection, valid IP address the app is tested to work fluently. With the recent advancements in technology, more reliable and convenient means of designing networks will be required. The aim of this project being an enterprise network, is to ensure that no device remains onto itself, there should be speed in the connectivity, addition of servers should not hinder the transfer of packets, added to that interfaces not meant to accesses should be blocked. It can be concluded that this aims were accomplished and totally completed to working and security standards.

Recommendations for Future Work

a) Additional access control lists (ACLs) should be implemented throughout the network to provide robust end-to-end security.

b) IPv6 addressing can be implemented to overcome any limitations in the number of hosts that can be used due to the available address space.

c) Additional encryption methods like RSA, SHA, SHA-2, and AES should be implemented to ensure security to client server's username and passwords.

We recommend every Network and Security enthusiasts to try this software and recommend if any changes required.

BIBILOGRAPHY

[1] Computer Network Demystified \. (2014). Retrieved December 12, 2014

[2] Djieva, P. T. *Introduction To Computer Networking.* Varna Free University,Institute of Technology.

[3] Easa, M. S. (n.d.). *CCNA in 21 Hours '640-802' Syllabus.* Retrieved from Bookboon.com: http://bookboon.com

[4] Enterprise Campus 3.0 Architecture:Overview and Framework . (n.d.). Retrieved November 11, 2014, from Cisco Networks: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/campover.html

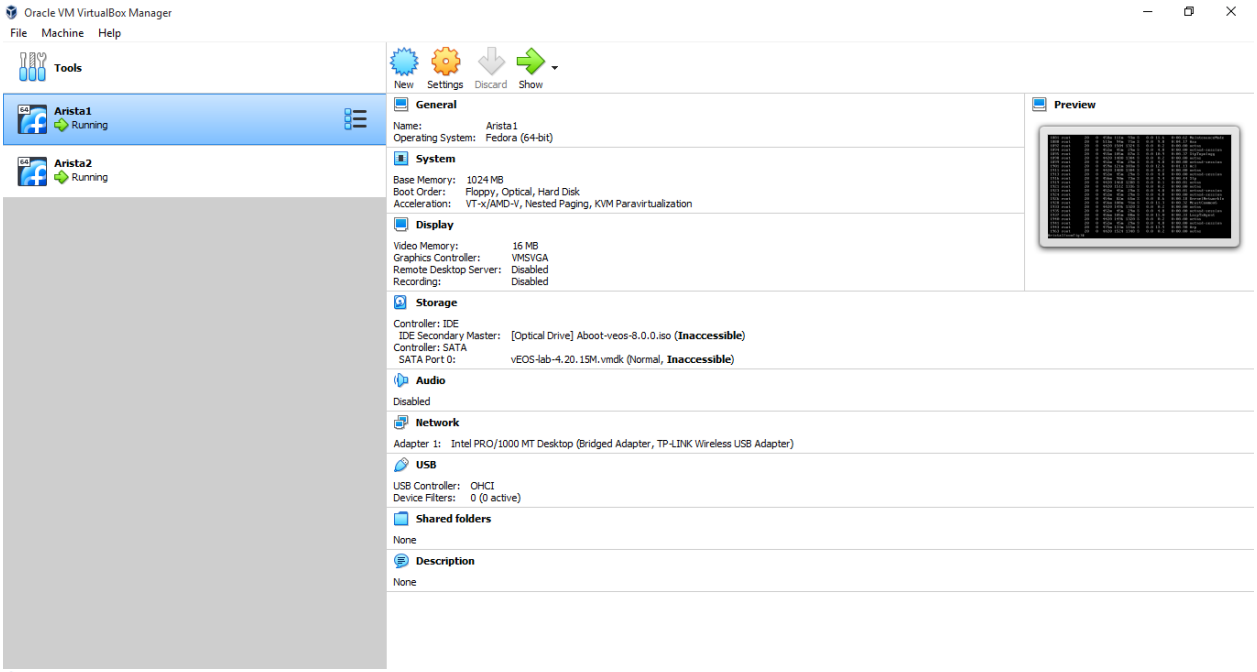
[5] Hierarchical Network Design-Layer of The Hierarchical Network Design Model. (n.d.). Retrieved August 17, 2014, from Cisco Networks: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/campover.html

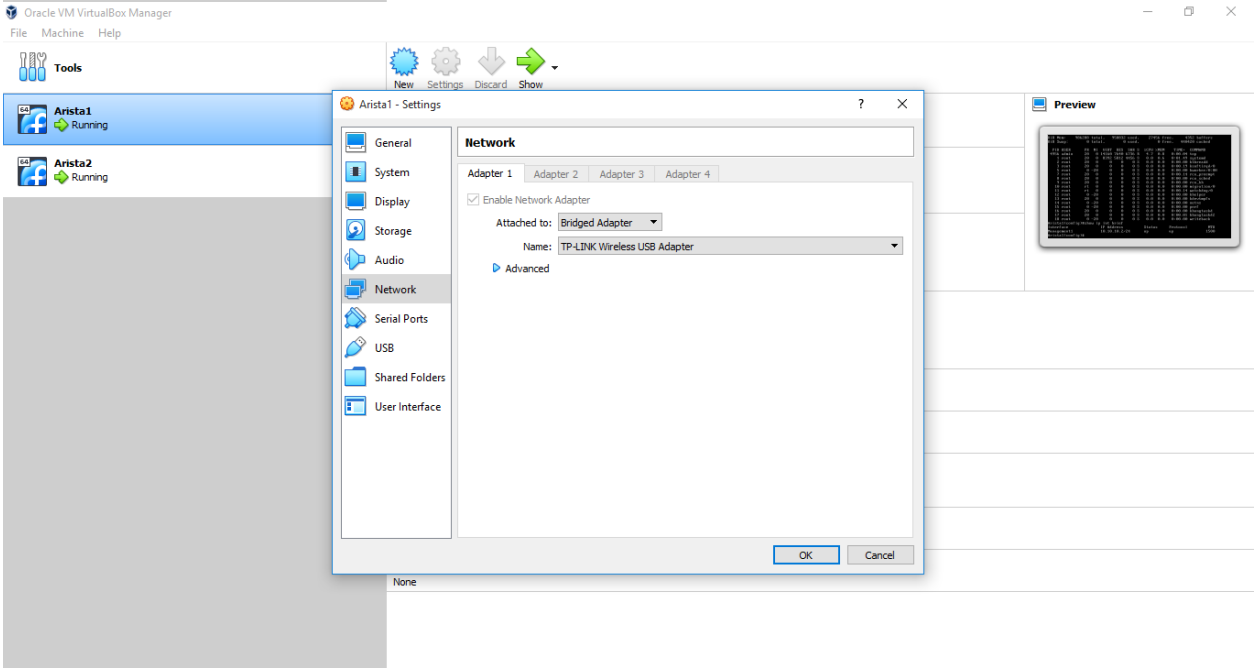
[6] Network Devices. (n.d.). Retrieved March 19, 2015, from Computer Networkig Notes: http://computernetworkingnotes.com/comptia-n-plus-study- guide/network-devices- hub-switch-router.html

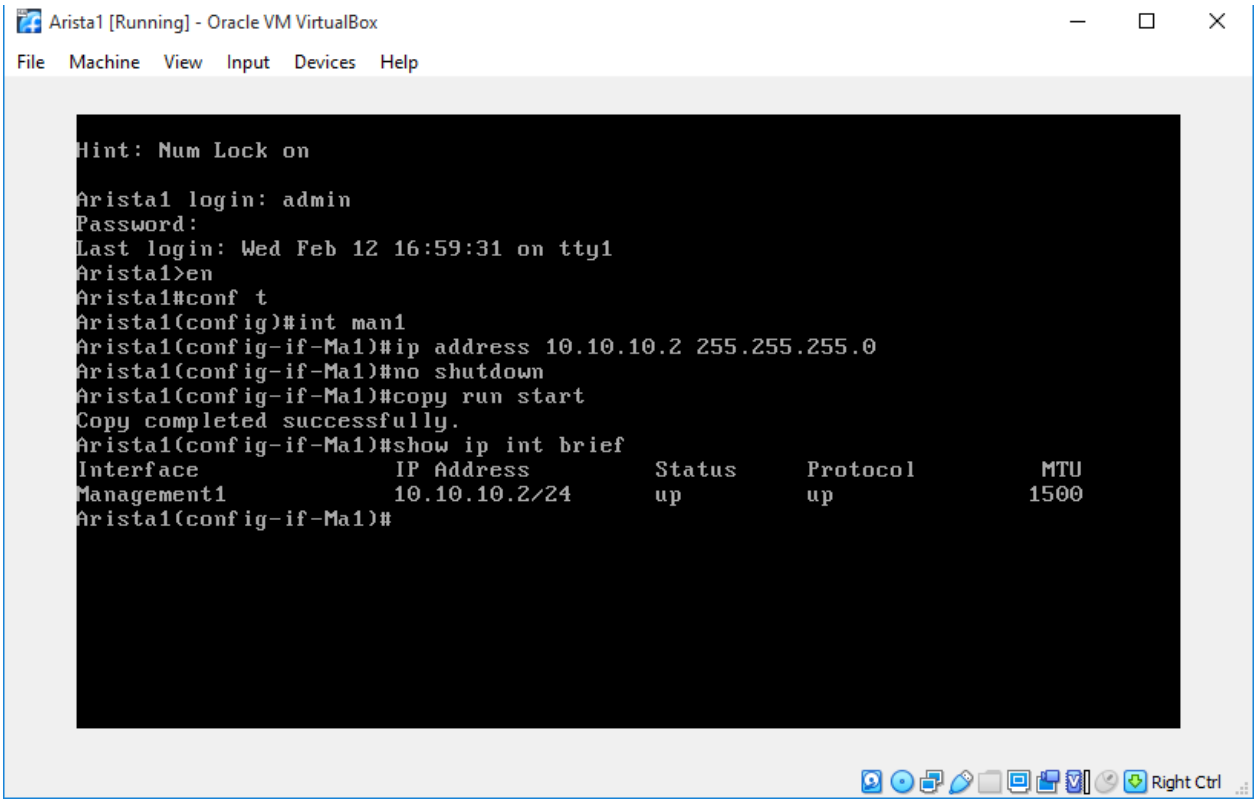
[7] Norberk Kircharians, P. P. (2014). *CCIE Routing and Switching V5.0 Official Cert Guide :IP Forwarding.*

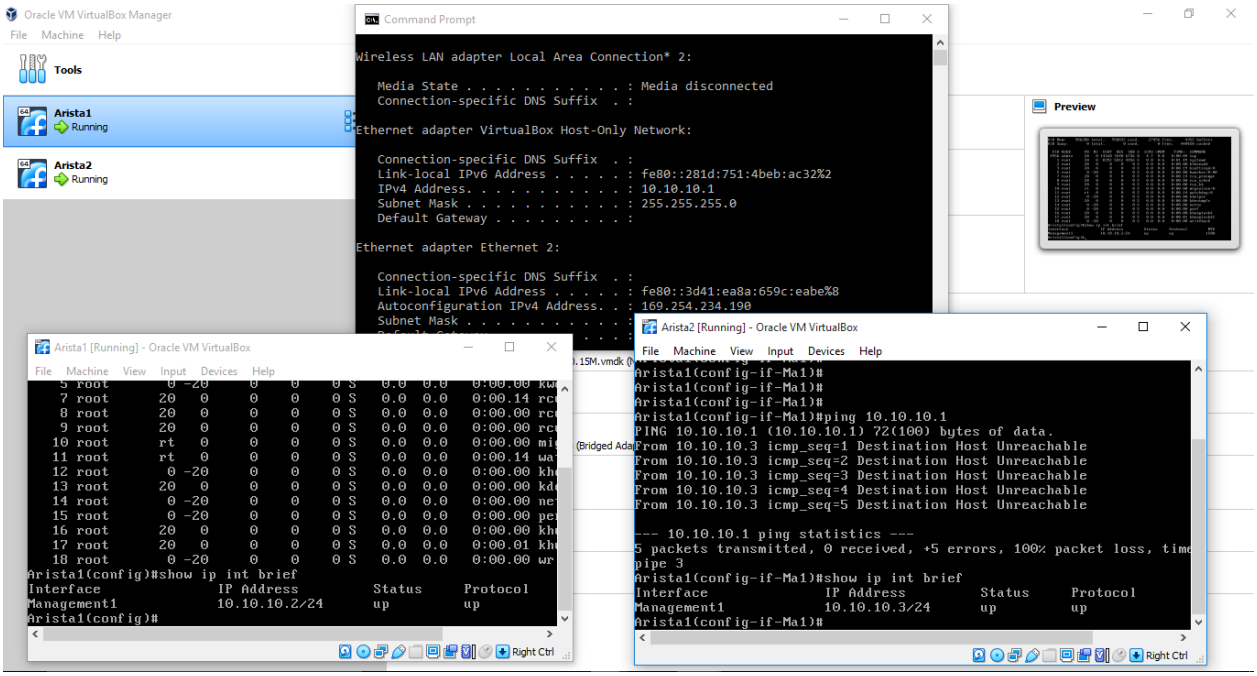
APPENDIX-1

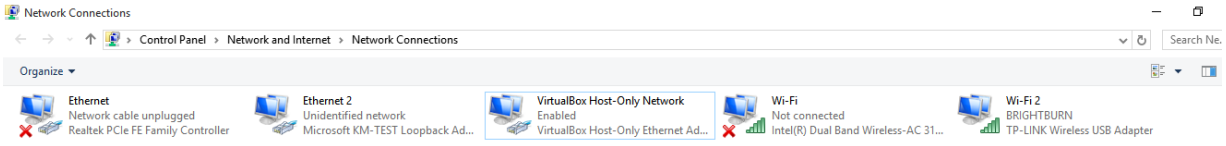
(Screenshots)

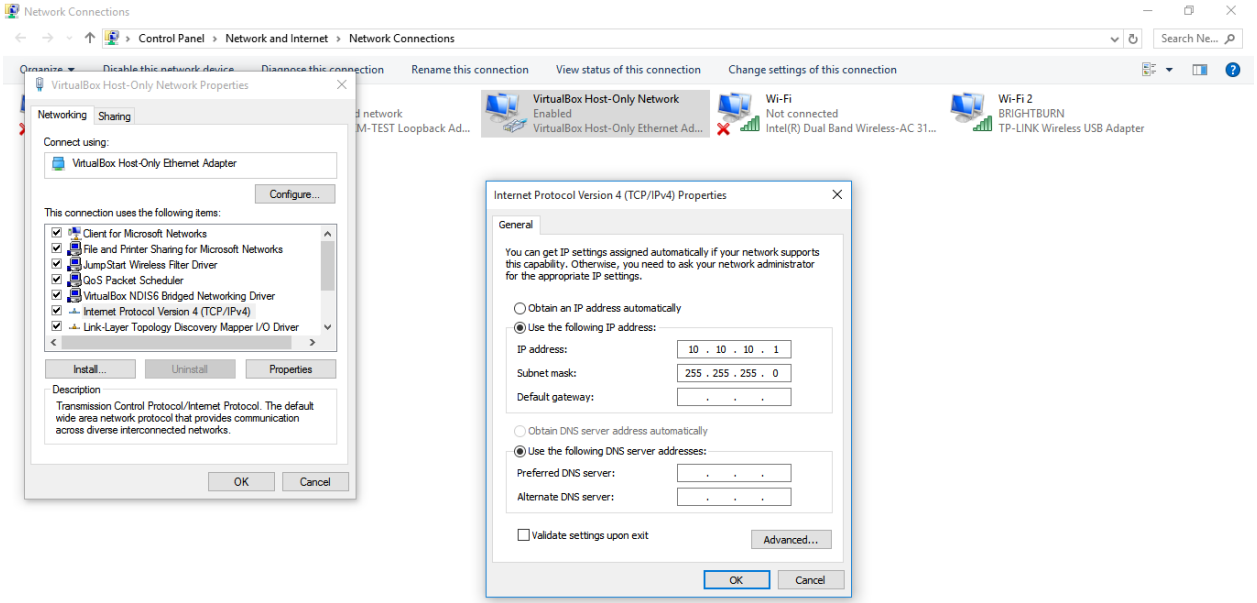


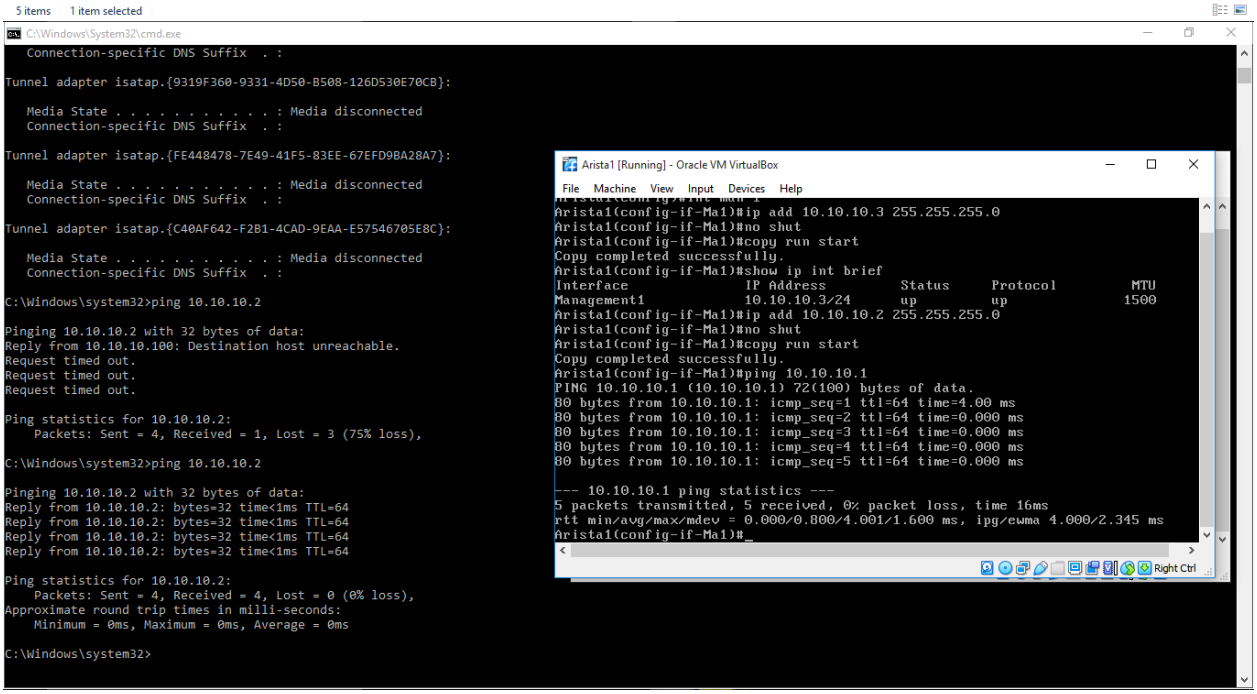


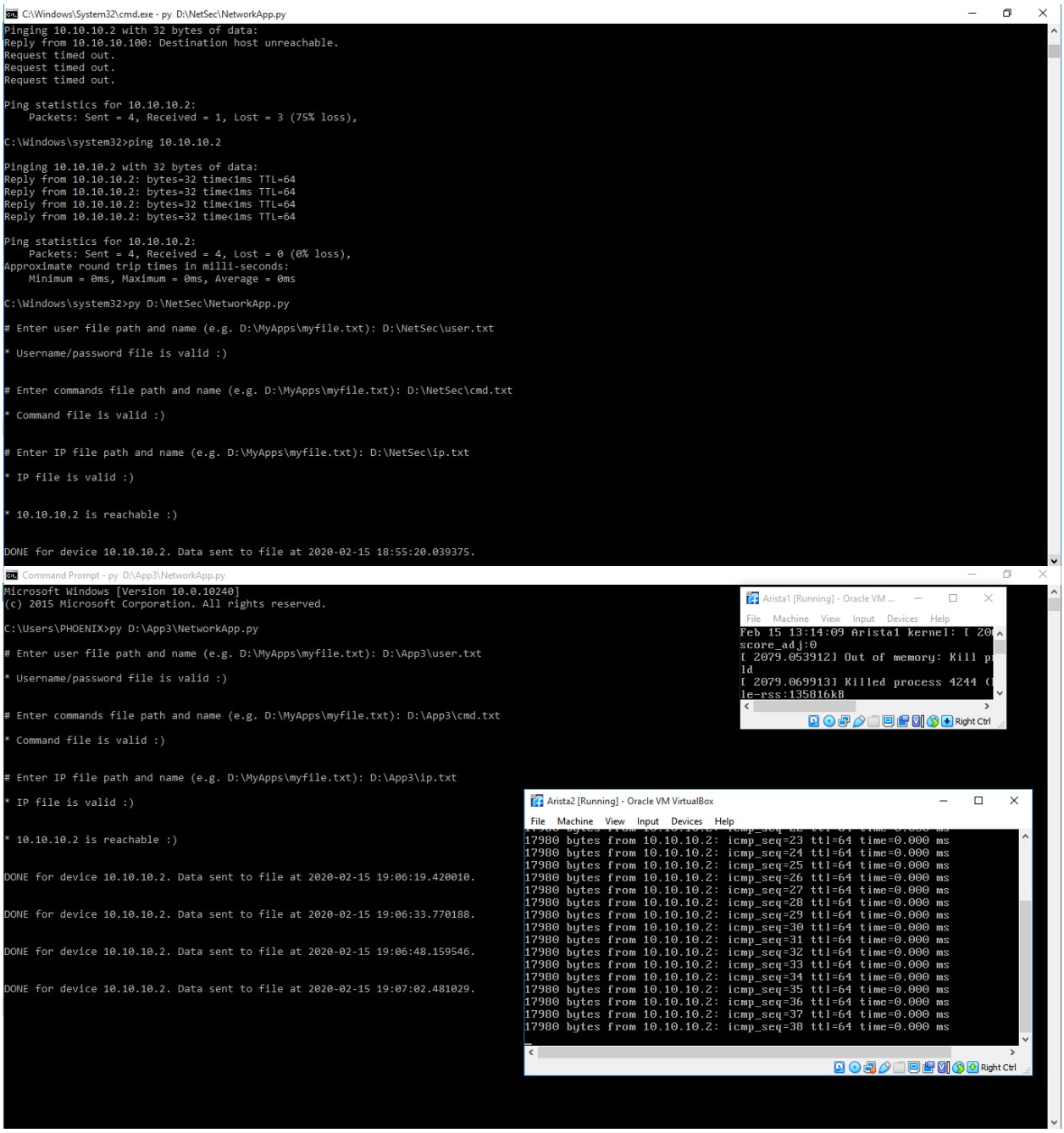

Arista1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Hint: Num Lock on
Arista1 login: admin
Password:
Last login: Wed Feb 12 16:59:31 on tty1
Arista1>en
Arista1#conf t
Arista1(config)#int man1
Arista1(config-if-Ma1)#ip address 10.10.10.2 255.255.255.0
Arista1(config-if-Ma1)#no shutdown
Arista1(config-if-Ma1)#copy run start
Copy completed successfully.
Arista1(config-if-Ma1)#show ip int brief
Interface IP Address Status Protocol MTU
Management1 10.10.10.2/24 up up 1500
Arista1(config-if-Ma1)#
Right Ctrl

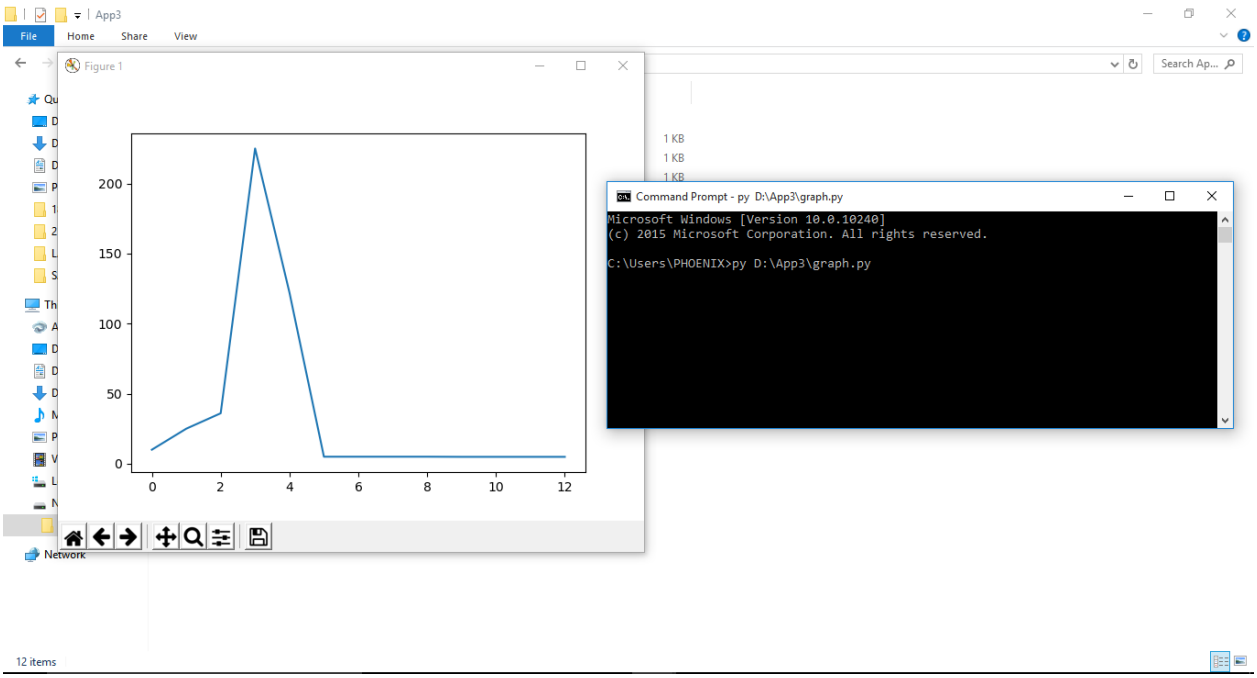

Oracle VM VirtualBox Manager
Tools
Arista1 Running
Arista2 Running
Command Prompt
Wireless LAN adapter Local Area Connection* 2:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Ethernet adapter VirtualBox Host-Only Network:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::281d:751:4beb:ac32%2
IPv4 Address. . . . . : 10.10.10.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
Ethernet adapter Ethernet 2:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::3d41:ea8a:659c:eabe%8
Autoconfiguration IPv4 Address. . : 169.254.234.190
Subnet Mask . . . . . :
Preview
Arista1 [Running] - Oracle VM VirtualBox
Arista1(config)#show ip int brief
Interface IP Address Status Protocol
Management1 10.10.10.2/24 up up
Arista1(config)#
Arista2 [Running] - Oracle VM VirtualBox
Arista1(config-if-Ma1)#ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 72(100) bytes of data.
From 10.10.10.3 icmp_seq=1 Destination Host Unreachable
From 10.10.10.3 icmp_seq=2 Destination Host Unreachable
From 10.10.10.3 icmp_seq=3 Destination Host Unreachable
From 10.10.10.3 icmp_seq=4 Destination Host Unreachable
From 10.10.10.3 icmp_seq=5 Destination Host Unreachable
--- 10.10.10.1 ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time
pipe 3
Arista1(config-if-Ma1)#show ip int brief
Interface IP Address Status Protocol
Management1 10.10.10.3/24 up up
Arista1(config-if-Ma1)#
Right Ctrl


Network Connections
Control Panel › Network and Internet › Network Connections
Organize
Ethernet
Network cable unplugged
Realtek PCIe FE Family Controller
Ethernet 2
Unidentified network
Microsoft KM-TEST Loopback Ad...
VirtualBox Host-Only Network
Enabled
VirtualBox Host-Only Ethernet Ad...
Wi-Fi
Not connected
Intel(R) Dual Band Wireless-AC 31...
Wi-Fi 2
BRIGHTBURN
TP-LINK Wireless USB Adapter


Network Connections
Control Panel › Network and Internet › Network Connections
Search Ne...
Rename this connection
View status of this connection
Change settings of this connection
VirtualBox Host-Only Network Properties
Networking
Sharing
Connect using:
VirtualBox Host-Only Ethernet Adapter
Configure...
This connection uses the following items:
Client for Microsoft Networks
File and Printer Sharing for Microsoft Networks
JumpStart Wireless Filter Driver
QoS Packet Scheduler
VirtualBox NDIS6 Bridged Networking Driver
Internet Protocol Version 4 (TCP/IPv4)
Link-Layer Topology Discovery Mapper I/O Driver
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
OK
Cancel
VirtualBox Host-Only Network
Enabled
VirtualBox Host-Only Ethernet Ad...
Wi-Fi
Not connected
Intel(R) Dual Band Wireless-AC 31...
Wi-Fi 2
BRIGHTBURN
TP-LINK Wireless USB Adapter
Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 10 . 10 . 10 . 1
Subnet mask: 255 . 255 . 255 . 0
Default gateway:
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server:
Alternate DNS server:
Validate settings upon exit
Advanced...
OK
Cancel


5 items
1 item selected
C:\Windows\System32\cmd.exe
Connection-specific DNS Suffix . :
Tunnel adapter isatap.{9319F360-9331-4D50-B508-126D530E70CB}:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Tunnel adapter isatap.{FE448478-7E49-41F5-83EE-67EFD9BA28A7}:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Tunnel adapter isatap.{C40AF642-F2B1-4CAD-9EAA-E57546705E8C}:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
C:\Windows\system32>ping 10.10.10.2
Pinging 10.10.10.2 with 32 bytes of data:
Reply from 10.10.10.100: Destination host unreachable.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.10.10.2:
Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
C:\Windows\system32>ping 10.10.10.2
Pinging 10.10.10.2 with 32 bytes of data:
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Ping statistics for 10.10.10.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Windows\system32>
Arista1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Arista1(config-if-Ma1)#ip add 10.10.10.3 255.255.255.0
Arista1(config-if-Ma1)#no shut
Arista1(config-if-Ma1)#copy run start
Copy completed successfully.
Arista1(config-if-Ma1)#show ip int brief
Interface IP Address Status Protocol MTU
Management1 10.10.10.3/24 up up 1500
Arista1(config-if-Ma1)#ip add 10.10.10.2 255.255.255.0
Arista1(config-if-Ma1)#no shut
Arista1(config-if-Ma1)#copy run start
Copy completed successfully.
Arista1(config-if-Ma1)#ping 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 72(100) bytes of data.
80 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=4.00 ms
80 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.000 ms
80 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.000 ms
80 bytes from 10.10.10.1: icmp_seq=4 ttl=64 time=0.000 ms
80 bytes from 10.10.10.1: icmp_seq=5 ttl=64 time=0.000 ms
--- 10.10.10.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 16ms
rtt min/avg/max/mdev = 0.000/0.800/4.001/1.600 ms, ipg/ewma 4.000/2.345 ms
Arista1(config-if-Ma1)#
Right Ctrl


C:\Windows\System32\cmd.exe - py D:\NetSec\NetworkApp.py
Pinging 10.10.10.2 with 32 bytes of data:
Reply from 10.10.10.100: Destination host unreachable.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 10.10.10.2:
Packets: Sent = 4, Received = 1, Lost = 3 (75% loss),
C:\Windows\system32>ping 10.10.10.2
Pinging 10.10.10.2 with 32 bytes of data:
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Reply from 10.10.10.2: bytes=32 time<1ms TTL=64
Ping statistics for 10.10.10.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Windows\system32>py D:\NetSec\NetworkApp.py
Enter user file path and name (e.g. D:\MyApps\myfile.txt): D:\NetSec\user.txt
* Username/password file is valid :)
# Enter commands file path and name (e.g. D:\MyApps\myfile.txt): D:\NetSec\cmd.txt
* Command file is valid :)
# Enter IP file path and name (e.g. D:\MyApps\myfile.txt): D:\NetSec\ip.txt
* IP file is valid :)
* 10.10.10.2 is reachable :)
DONE for device 10.10.10.2. Data sent to file at 2020-02-15 18:55:20.039375.
Command Prompt - py D:\App3\NetworkApp.py
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\PHOENIX>py D:\App3\NetworkApp.py
Enter user file path and name (e.g. D:\MyApps\myfile.txt): D:\App3\user.txt
* Username/password file is valid :)
# Enter commands file path and name (e.g. D:\MyApps\myfile.txt): D:\App3\cmd.txt
* Command file is valid :)
# Enter IP file path and name (e.g. D:\MyApps\myfile.txt): D:\App3\ip.txt
* IP file is valid :)
* 10.10.10.2 is reachable :)
DONE for device 10.10.10.2. Data sent to file at 2020-02-15 19:06:19.420010.
DONE for device 10.10.10.2. Data sent to file at 2020-02-15 19:06:33.770188.
DONE for device 10.10.10.2. Data sent to file at 2020-02-15 19:06:48.159546.
DONE for device 10.10.10.2. Data sent to file at 2020-02-15 19:07:02.481029.
Arista1 [Running] - Oracle VM ...
File Machine View Input Devices Help
Feb 15 13:14:09 Arista1 kernel: [20
score_adj:0
[2079.053912] Out of memory: Kill p
ld
[2079.069913] Killed process 4244 (
le-rss:135816kB
Right Ctrl
Arista2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
17980 bytes from 10.10.10.2: icmp_seq=23 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=24 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=25 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=26 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=27 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=28 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=29 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=30 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=31 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=32 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=33 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=34 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=35 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=36 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=37 ttl=64 time=0.000 ms
17980 bytes from 10.10.10.2: icmp_seq=38 ttl=64 time=0.000 ms
Right Ctrl


App3
File
Home
Share
View
Search Ap...
Figure 1
200
150
100
50
0
0
2
4
6
8
10
12
Command Prompt - py D:\App3\graph.py
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\PHOENIX>py D:\App3\graph.py
Network
12 items

Appendix-2
(Codes)

Configuring swicthes

Arista1

enable
configure terminal
username admin secret python
hostname Arista1
interface Management 1
ip address 10.10.10.2 255.255.255.0
no shutdown
copy run start

Arista2

enable
configure terminal
username admin secret python
hostname Arista2
interface Management 1
ip address 10.10.10.3 255.255.255.0
no shutdown
copy run start

#1 Checking IP address file and content validity

```
import os.path
import sys

def ip_file_valid():

   #Prompting user for input
   ip_file = input("\n# Enter IP file path and name (e.g. D:\MyApps\myfile.txt): ")

   #Changing exception message
   if os.path.isfile(ip_file) == True:
      print("\n* IP file is valid :)\n")

   else:
      print("\n* File {} does not exist :( Please check and try again.\n".format(ip_file))
      sys.exit()

   #Open user selected file for reading (IP addresses file)
   selected_ip_file = open(ip_file, 'r')

   #Starting from the beginning of the file
   selected_ip_file.seek(0)
```

```
   #Reading each line (IP address) in the file
   ip_list = selected_ip_file.readlines()

   #Closing the file
   selected_ip_file.close()

   return ip_list
```

#2 Checking IP address validity

```
import sys

#Checking octets
def ip_addr_valid(list):

   for ip in list:
      ip = ip.rstrip("\n")
      octet_list = ip.split('.')

      if (len(octet_list) == 4) and (1 <= int(octet_list[0]) <= 223) and (int(octet_list[0]) != 127) and (int(octet_list[0]) != 169 or int(octet_list[1]) != 254) and (0 <= int(octet_list[1]) <= 255 and 0 <= int(octet_list[2]) <= 255 and 0 <= int(octet_list[3]) <= 255):
         continue

      else:
         print('\n* There was an invalid IP address in the file: {} :(\n'.format(ip))
         sys.exit()
```

#3Checking IP address reachability

```
import sys
import subprocess

#Checking IP reachability
def ip_reach(list):

   for ip in list:
      ip = ip.rstrip("\n")

      ping_reply = subprocess.call('ping %s -n 2' % (ip,), stdout=subprocess.DEVNULL, stderr=subprocess.DEVNULL)

      if ping_reply == 0:
         print("\n* {} is reachable :)\n".format(ip))
         continue

      else:
         print('\n* {} not reachable :( Check connectivity and try again.'.format(ip))
         sys.exit()
```

#4 Creating SSH connection

```
import paramiko
import datetime
import os.path
import time
```

```
import sys
import re

#Checking username/password file
#Prompting user for input - USERNAME/PASSWORD FILE
user_file = input("\n# Enter user file path and name: ")

#Verifying the validity of the USERNAME/PASSWORD file
if os.path.isfile(user_file) == True:
   print("\n* Username/password file is valid :)\n")

else:
   print("\n* File {} does not exist :( Please check and try again.\n".format(user_file))
   sys.exit()

#Checking commands file
#Prompting user for input - COMMANDS FILE
cmd_file = input("\n# Enter commands file path and name: ")

#Verifying the validity of the COMMANDS FILE
if os.path.isfile(cmd_file) == True:
   print("\n* Command file is valid :)\n")

else:
   print("\n* File {} does not exist :( Please check and try again.\n".format(cmd_file))
   sys.exit()

#Open SSHv2 connection to the device
def ssh_connection(ip):

   global user_file
   global cmd_file

   #Creating SSH CONNECTION
   try:
      #Define SSH parameters
      selected_user_file = open(user_file, 'r')

      #Starting from the beginning of the file
      selected_user_file.seek(0)

      #Reading the username from the file
      username = selected_user_file.readlines()[0].split(',')[0].rstrip("\n")

      #Starting from the beginning of the file
      selected_user_file.seek(0)

      #Reading the password from the file
      password = selected_user_file.readlines()[0].split(',')[1].rstrip("\n")

      #Logging into device
      session = paramiko.SSHClient()

      session.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
    #Connect to the device using username and password
    session.connect(ip.rstrip("\n"), username = username, password = password)

    #Start an interactive shell session on the router
    connection = session.invoke_shell()

    #Setting terminal length for entire output
    connection.send("enable\n")
    connection.send("terminal length 0\n")
    time.sleep(1)

    #Entering global config mode
    connection.send("\n")
    connection.send("configure terminal\n")
    time.sleep(1)

    #Open user selected file for reading
    selected_cmd_file = open(cmd_file, 'r')

    #Starting from the beginning of the file
    selected_cmd_file.seek(0)

    #Writing each line in the file to the device
    for each_line in selected_cmd_file.readlines():
       connection.send(each_line + '\n')
       time.sleep(2)

    #Closing the user file
    selected_user_file.close()

    #Closing the command file
    selected_cmd_file.close()

    #Checking command output for IOS syntax errors
    router_output = connection.recv(65535)

    if re.search(b"% Invalid input", router_output):
       print("* There was at least one IOS syntax error on device {} :(".format(ip))

    else:
       print("\nDONE for device {}. Data sent to file at {}.\n".format(ip, str(datetime.datetime.now())))

    #Test for reading command output
    #print(str(router_output) + "\n")

    #Searching for the CPU utilization value within the output of "show processes top once"
    cpu = re.search(b"%Cpu\(s\):(\s)+(.+?)(\s)* us,", router_output)

    #Extracting the second group, which matches the actual value of the CPU utilization and decoding to the UTF-8
format from the binary data type
    utilization = cpu.group(2).decode("utf-8")

    #Printing the CPU utilization value to the screen
```

```
        #print(utilization)

        #Opening the CPU utilization text file and appending the results
        with open("D:\\App3\\cpu.txt", "a") as f:
            #f.write("{},{}\n".format(str(datetime.datetime.now()), utilization))
            f.write(utilization + "\n")

        #Closing the connection
        session.close()

    except paramiko.AuthenticationException:
        print("* Invalid username or password :( \n* Please check the username/password file or the device configuration.")
        print("* Closing program... Bye!")
```

#5 creating graph from captured packets

```
import matplotlib.pyplot as pyp
import matplotlib.animation as animation

#Creating a new figure
figure = pyp.figure()

#Creating a subplot with 1 row, 1 column and index 1 - this means a single subplot in our figure
subplot = figure.add_subplot(1, 1, 1)

#Creating the function that reads the data from cpu.txt and feeds it to our subplot
def animation_function(i):
    #Opening the file and reading each row of CPU utilization data in the file; creating a list of values
    cpu_data = open("D:\\App3\\cpu.txt").readlines()

    #Creating an empty list in which to append each value in the file converted from string to float;
    x = []

    #Iterating over the list of CPU values and appending each value (converted to float) to the previously created list - x; adding an if statement to exclude any blank lines in the file
    for each_value in cpu_data:
        if len(each_value) > 1:
            x.append(float(each_value))

    #Clearing/refreshing the figure to avoid unnecessary overwriting for each new poll (every 10 seconds)
    subplot.clear()

    #Plotting the values in the list
    subplot.plot(x)
#Using the figure, the function and a polling interval of 10000 ms (10 seconds) to build the graph
graph_animation = animation.FuncAnimation(figure, animation_function, interval = 10000)

#Displaying the graph to the screen
pyp.show()
```