المجلة الإلكترونية الشاملة متعددة التخصصات
Electronic Interdisciplinary Miscellaneous Journal
العدد التاسع والستين شهر (3) 2024
Issue 69, (3) 2024
ISSN: 2617-958X

RsTool: Self-healing Network Tool for Restarting Server: Practical Experiment for Developing Network Infrastructure Monitoring

Author: Fahad Mohammed Al-Mohaimeed [1]

[1] Computer Network Specialist, Ministry of Health, Kingdom of Saudi Arabia, Jeddah,
fmalmohaimeed@moh.gov.sa

Abstract

Considering the developments taking place in the field of information technology and in computer networks, and the rapid changes in the use and dependence of applications and programs on a large scale, and after what also happened in the past years from the effects in the world due to Corona pandemic, resulting in an increase in the use of the network, thus increasing the density of mobile data across the network. The network self-healing method in such cases is very useful as it handles errors that occur or network failures. This paper reviews an idea of developing an automatic server restart tool, the server responsible for allocating network addresses in case of failure, and the methodology used to develop this tool and its benefits for the facility's network.

Keywords: Self-healing network, Network Automation, Network Server, Ping message command, Python.

ملخص البحث

بالنظر إلى التطورات التي تحدث في مجال تكنولوجيا المعلومات وشبكات الحاسوب، والتغيرات السريعة في استخدام التطبيقات والبرامج والاعتماد عليها على نطاق واسع، وبعد ما حدث أيضًا في السنوات الماضية من تأثيرات في العالم بسبب جائحة كورونا، مما أدى إلى زيادة استخدام الشبكة والاعتماد عليها، وبالتالي زيادة كثافة البيانات المتنقلة عبر الشبكة. تعد طريقة الإصلاح الذاتي للشبكة (أو المعالجة الذاتية للشبكة) في مثل هذه الحالات مفيدة جدًا لأنها تعالج الأخطاء التي تحدث في الشبكة أو فشل الشبكة. تستعرض هذه الورقة فكرة تطوير أداة إعادة التشغيل التلقائي للخادم، والخادم المسؤول عن تخصيص عناوين الشبكة في حالة الفشل، والمنهجية المستخدمة لتطوير هذه الأداة وفوائدها لشبكة المنشأة.

الكلمات المفتاحية: الإصلاح الذاتي للشبكة، أتمتة الشبكة، خادم الشبكة، أمر رسالة بينج، بايثون.

Introduction

Self-healing network is one of the most important measures taken in network management and development, especially with the increasing use of the network. It is a method to manage the network with human less and solve issues that may occurred in the network. This method may be a tool or software, it often performs a corrective action on the network automatically after alerting the network administrator or sending a message explaining the action that will be applied for the self-healing operation [1].

With increase in the use of Internet (the global network) around the world and in establishments in particular, network disruption often occurs, and this was observed through past events related to the Covid pandemic, and even after its relative decline in some countries, the dependence on the use of the network is still increasing, so the importance of the self-treatment network lies here in Business continuity and reducing disruption.

The use of programming in developing tools for self-healing network is an important issue within every organization. This paper aims to provide a tool that has been developed to automatically restart the server responsible for distributing network addresses to users' devices in the event of its failure.

Research Problem

Business continuity is needed Even if there is a network malfunction, this risk must be reduced in order to ensure business continuity within the facility. Imagine if the facility is a hospital. Disruption of the network in this scenario is considered relatively dangerous. Emergency procedures, the laboratory, or even the pharmacy may be disrupted, causing congestion of people waiting and the disruption of some health measures taken, which poses a threat to people's health. It is necessary to monitor the operation of the network. A tool was developed using Python that monitors the server responsible for distributing network addresses within the hospital and restarts it when the network stops working.

Since the computer network has become the backbone of every facility or organization, the importance of the network’s continuous work within the health facility in particular, in order to provide technical services on an ongoing basis to both the patient and the treating physician, in order to facilitate procedures from the moment the patient enters the hospital gate until he leaves the hospital. So, the goal here is to enable the network to quickly respond to any issue that occurs in it to ensure business continuity in these facilities.

Research Objectives

Self-healing network in general aims to reducing network failure and increasing quick responding to issue that occur in the network [1]. And from [2] the benefits of self-healing network are mentioned in general with an emphasis on the goal of maintaining stability and security. Depends on that the objectives of using RsTool within facility, especially health care, are:

1- Reducing human failure: by reducing human interventions that sometimes causes a waste of time to solve issues related to the network and the reason that the staff of network department may not be present due to working hours, and network disruption may occur at any time, especially in health facilities that require the network to operate over time. So, checking the network and restarting server producers works automatically as RsTool, it gives better result.

2- Business continuity: by making the network do self-restarting using RsTool. So, working procedure within the facility continuous, and even there is an interruption it will be slight.

3- Risk recovery: if network failure happens, the recovery by the tool is done by performing server restarting action automatically.

4- Saving time: this is by acting by the tool without to involve human to do restarting process or healing the network. Also, depending on the results, the duration of the network disruption decreased from 30 minutes to 10 minutes. Therefore, if the facility is a health facility, it will result in reducing the waiting time for patients when the network is disruption.

5- Guarantee network security: through developing RsTool locally and full control from inside rather than from outside with authentication procedures access to the tool.

Methodology

RsTool is a software agent for self-healing network, developed by using Python programming language with console based. Also, this is a form of network automation. The tool works on restarting process for the server that responsible for allocating network address. Python script here takes the healing action [3].

First, remotely restarting procedure in the operating system (Microsoft Windows in this research paper) has been enabled that related to the server so that it receives this action from the device designated for managing the network. After running the tool on the device designated for managing the network, the user who is the network administrator is asked how the remotely restarting procedure works, whether it is based on the network address related to the server or the name of the server, then asks to enter the number of times the network has been stopped, if it is the first time the tool is running, the number of times will be zero. Otherwise, the network administrator will be advised to verify the number of times via e-mail. After that, tool starts executing the ping command (sending data) twice to Server IP address to make sure that the network is working or not. If one of these messages is not cycled twice, it means that the devices are not connected to the Internet and a remote reboot of the server must be performed. During this, an email is sent to the network administrator that the reboot procedure will be started for the server within seconds with the date and time, then the command is sent to the server to do the restart. Then the tool puts itself in sleep mode for a period of 10 minutes allow the server to work again and perform the network address allocation process. **time.sleep()** function in python does insert time delay to give more time to make sure the network and server are up and running after restarting procedure [4]. The following figure shows architecture of RsTool:

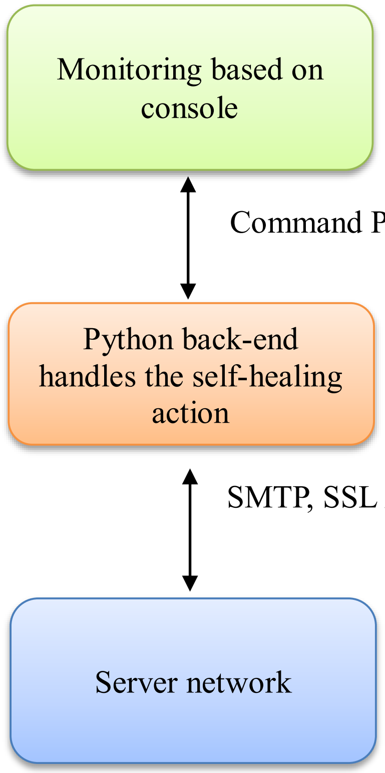

Monitoring based on console
Command Prompt
Python back-end handles the self-healing action
SMTP, SSL / Win32com.client
Server network

Figure 1: Architecture of RsTool.

The following figure describes the flow chart diagram of the tool after start working after granting access permissions:

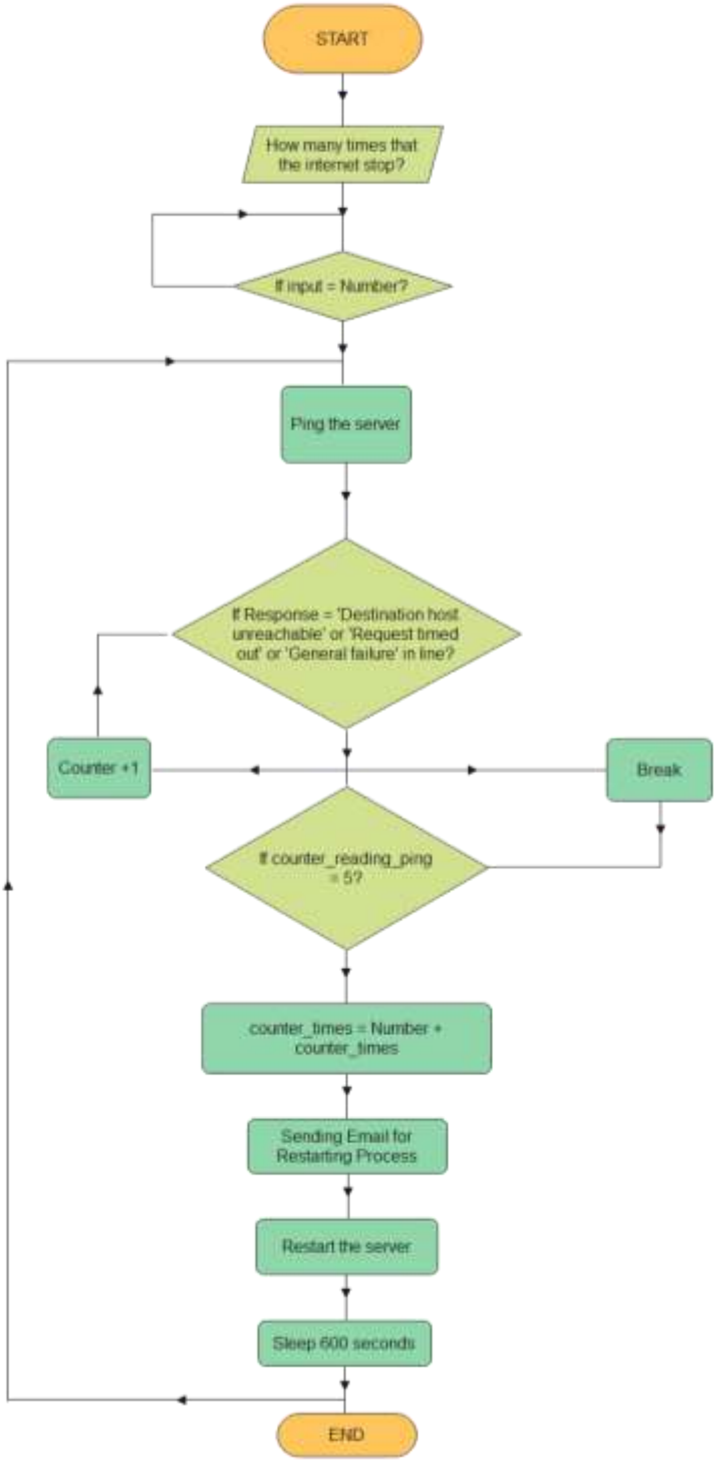

START
How many times that the internet stop?
If input = Number?
Ping the server
If Response = 'Destination host unreachable' or 'Request timed out' or 'General failure' in line?
Counter +1
Break
If counter_reading_ping = 5?
counter_times = Number + counter_times
Sending Email for Restarting Process
Restart the server
Sleep 600 seconds
END

Figure 2: Flowchart diagram of RsTool.

After running the tool there is an email message will be sent to the network administrator contains of a temporary password and information about the time and date of operation of the tool. The email is using outlook application with **win32com.client** module. The temporary password was created by password generator function using random module with shuffle method **random.shuffle()** and random character format **"!@#$%^&*()"**. The complete function will be as in the following figure code in Python:

```
def generate_random_password(x):
    characters = list(string.ascii_letters + string.digits +
    "!@#$%^&*()")
    length = x
    random.shuffle(characters)
    password = []
    for i in range(length):
        password.append(random.choice(characters))
    random.shuffle(password)
    password_join = "".join(password)
    date = Time.strftime("%Y-%m-%d")
    date_str = str(date) + " "
    time_ = Time.strftime("%H:%M %p")
    password_str = str(password_join)
    outlook = win32com.client.Dispatch('outlook.application')
    mail = outlook.CreateItem(0)
    mail.To = 'network_admin@mail.com'
    mail.Subject = 'Email from Network'
    mail.HTMLBody = '<h2>This is HTML Body</h2>'
    mail.Body = "The tool has been run, this is the password
    to continue "  + password_str + " and time " + time_
    mail.Send()
```

Figure 3: Shows Python code for function of generating random password.

After that, Graphical User Interface - GUI **Tkinter** window opens with size 550x450 to enter the password sent by the email from the generator function to start using the tool. The mechanism for reading ping and its results was using the **ImportOS** module, as in [5], and saving the reading result in a variable called **Response**, but in this research several nested loops were used, containing reading the time since the tool started the actual work of the ping command process, and defining a counter to count the number of times the ping was read and showed a result for non-response. If the timer is five times, an email is sent to the network administrator to inform that this server is not responding and a reboot procedure must be initiated, along with sending the time and date. The following image is showing the code for asking the network administrator to do remote restart with name (write 1) or IP address (write 2)? after that will enter the host name or the host IP (**int(result)** method for convert String inputs to integer and save it in **result_in variable**).

```
Result = input("Remote restart (by name or/ by IP)? (1 / 2):")
result_int = int(result)
if result_int == 1:
    name = input("Enter host name: ")
elif result_int == 2:
    ip = input("Enter host IP: ")
```

Figure 4: Shows Python code for the choices between remote restart with hot name or IP address.

In **While True** below starting the counter value in 1, and then the ping process repeated twice in **range(2)** and save the result in **response, splitlines()** method is applied five times and it used for to split string into lists to make comparison to the response generated by ping command process. The time and date is saved into **Time** variable. If the counter value reaches 5 that means the response from ping command is negative "Destination host unreachable, Request timed out or General failure". After the email is sent the tool goes to sleep mode for 10 minutes (600 seconds) until start again. After this practical experiment, 10 minutes is an appropriate time for solving the network issue and beginning the restart process, the experiment took place in a health facility.

```
while True:
    counter = 1
    for i in range(2):
        counter = 1
        response = os.popen(f"ping {ip_server}").read()
        for i in range(1, 5):
            line = response.splitlines()[i + 1]
            if ": Destination host unreachable." in line:
                counter = counter + 1
            elif " Request timed out." in line:
                counter = counter + 1
            elif " General failure." in line:
                counter = counter + 1
            else:
                quit()
            Time = datetime.datetime.now()

    if (counter == 5):
        print("Ping unsuccessful")
        date = Time.strftime("%Y-%m-%d")
        time_ = Time.strftime("%H:%M %p")
        print(date)
        print(time_)
        email_sending_counter = email_sending_counter + 1
        outlook = win32com.client.Dispatch
        ('outlook.application')
        mail = outlook.CreateItem(0)
        mail.To = 'network_admin@mail.com'
        mail.Subject = 'Email from Network'

        mail.HTMLBody = '<h2>This is HTML Body</h2>'
        email_sending_counter_int = int(email_sending_
        counter)
        email_sending_counter_str = str(email_sending_
        counter_int)
        mail.Body = "The network has an issue: restarting
        process for the server is running now in"
        + date + " & time " + time_ + ". This is the "
        + email_sending_counter_str + " time that the
        Internet has stopped."
        mail.Send()
        print("Restart host ip: ", ip)
        os.system("shutdown /m " + ip + " /r /f")
    else:
        print("Ping reading successful")
    time.sleep(600)
```

Figure 5: Shows Python code for the back-end self-healing action (restarting process).

Results & Images

The results showed that RsTool to do restarting process succussed, and an email message was sent to network administrator, the message contains of times and date that the internet has stopped and numbers of times of that the restarting process has been done. The experiment was on two devices, one device with RsTool installed on it, and the other device (the server) receives restarting command from the first device.

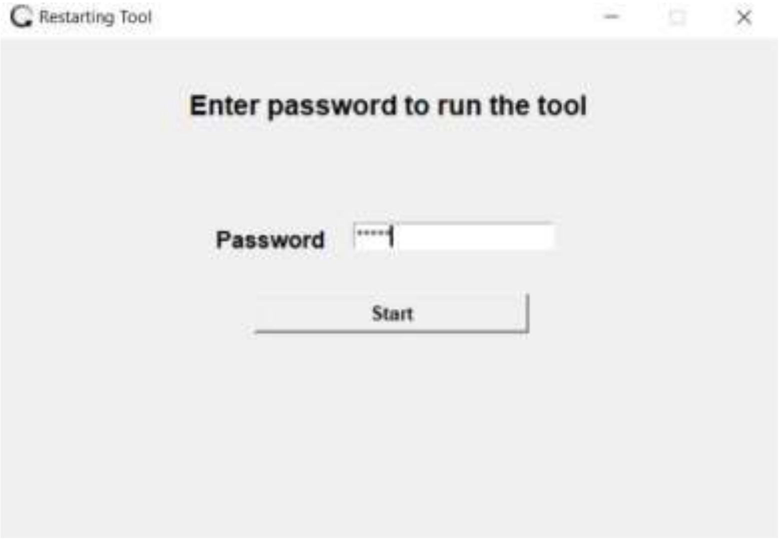
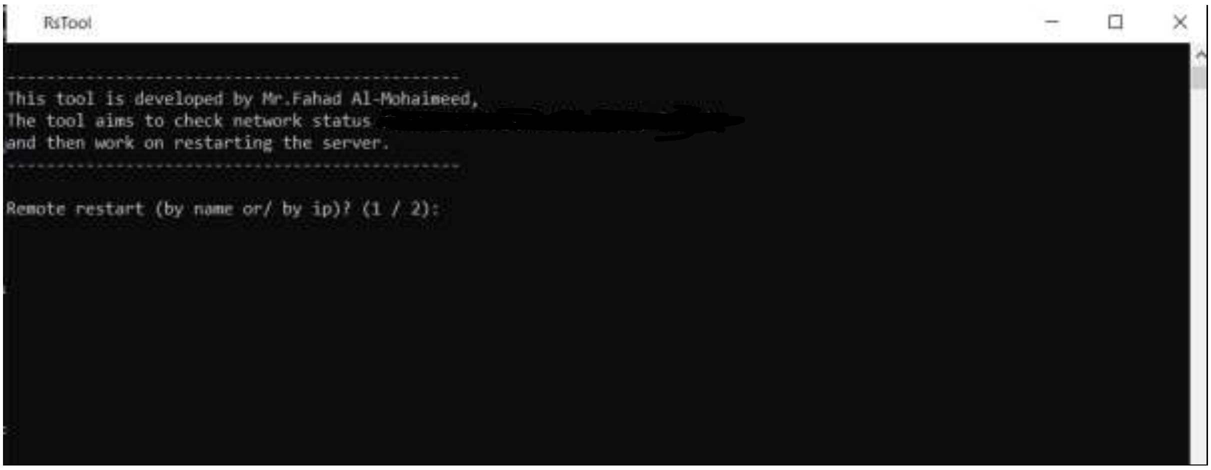


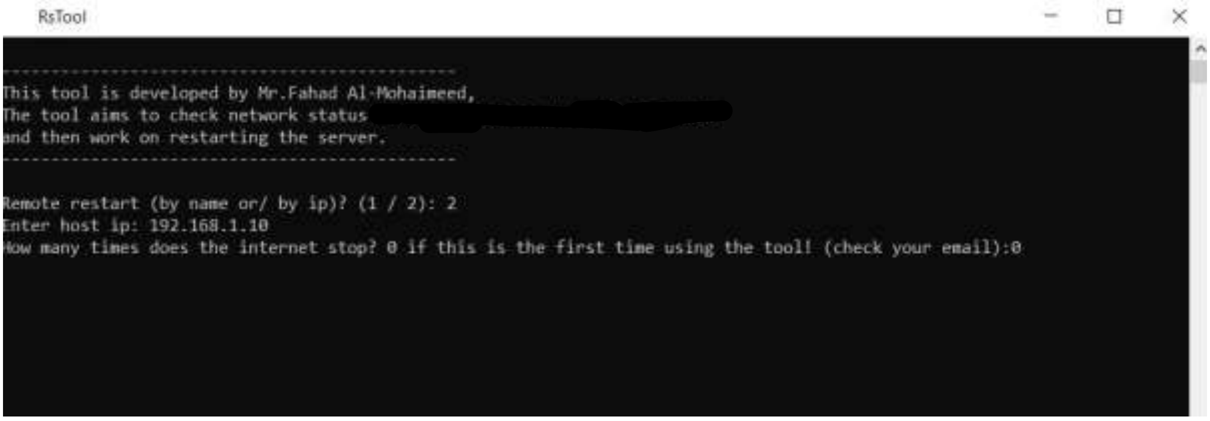
Figure 6: Shows the starting window for the tool.



```
----------------------------------------------
This tool is developed by Mr.Fahad Al-Mohaimeed,
The tool aims to check network status [redacted]
and then work on restarting the server.
----------------------------------------------

Remote restart (by name or/ by ip)? (1 / 2):
```

Figure 7: Shows that RsTool ask the user to do remote restart by name or by IP address after running the tool.



```
RsTool

----------------------------------------------------
This tool is developed by Mr.Fahad Al-Mohaimeed,
The tool aims to check network status [redacted]
and then work on restarting the server.
----------------------------------------------------

Remote restart (by name or/ by ip)? (1 / 2): 2
Enter host ip: 192.168.1.10
How many times does the internet stop? 0 if this is the first time using the tool! (check your email):0
```

Figure 8: Shows the RsTool ask the network administrator to enter IP address of the server that should be restarted and enter times the internet has stopped.

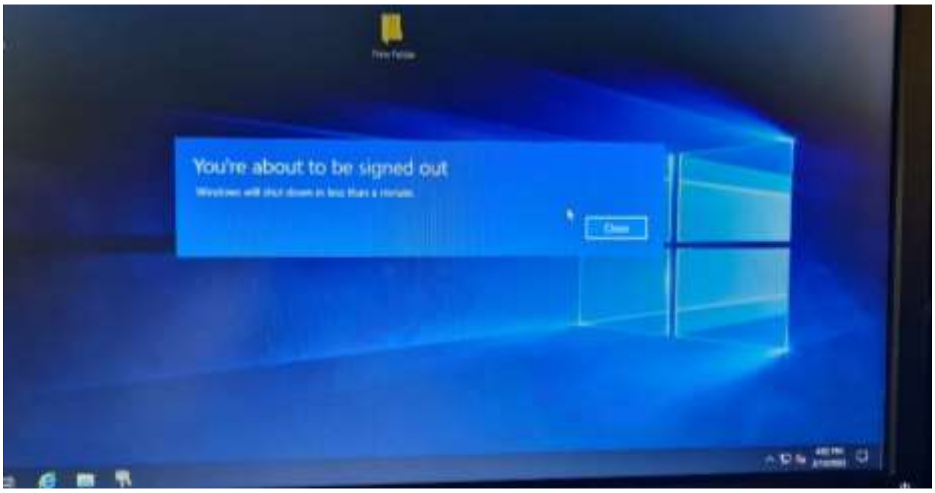
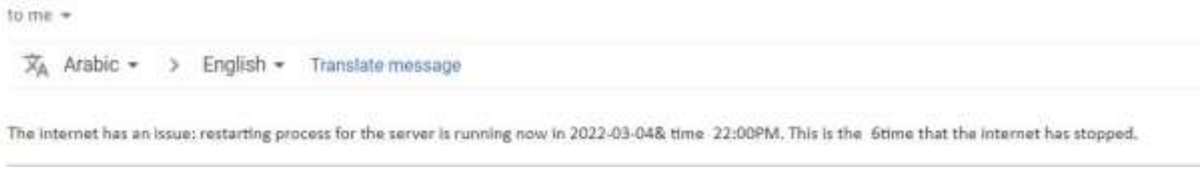


Figure 9: Shows the restarting procedure will start after sending the restart command from the tool in the sender device to receiver server.



The internet has an issue: restarting process for the server is running now in 2022-03-04& time 22:00PM. This is the 6time that the internet has stopped.

Figure 10: Shows an email message sent from the tool to network administrator.

Conclusion & Recommendations

The results were highly successful, as the tool will be adopted to manage and monitor the network, and based on these results, the goals of this tool will be achieved. This paper also recommends the importance of finding possible solutions with regard to the facility's network, in order to protect against possible malfunctions. And software solutions related to the network are considered among the modern and advanced topics at this time, as they are characterized by the speed of completion, and this is what is required for a quick technical solution.

References

[1] Help Systems. (2017, August 19). The Benefits of a Self-Healing Network. Retrieved from https://www.helpsystems.com/blog/benefits-self-healing-network

[2] IBM. (2021, May 16). How self-healing networks help keep digital world and secure. Retrieved from https://www.ibm.com/blogs/industries/self-healing-networks-stable-secure-digital-5g-world/

[3] Kapadia, Krishma & Maity, Sangita & Shari, Shikna & Bhowmik, Soumita & Deshpande, Tanvi & DR. Gifford, Kevin - Faculty Advisor & DR. Perigo, Levi - Academic Advisor & Xu, Zhenggen – Industry Advisor. (2018) Self-Healing Network System (SHNS), Interdisciplinary Telcom Program University of Colorado Boulder, 2018.

[4] Chou, E. (2017). Mastering Python Networking. Packet Publishing Ltd, 2017.

[5] Ronnie Atuhaire's Blog. (2022, April 18), Ping with Python. Retrieved from https://blog.octachart.com/ping-with-python#heading-os-module