

# Downtime Tracker

Jerried Ramon, Derek Williams\*

Department of Computer Science  
Florida Polytechnic University  
4700 Research Way, Lakeland,  
FL-33805, USA  
[derekwilliams2002@flpoly.org](mailto:derekwilliams2002@flpoly.org)

**Abstract** - Websites and online service cannot always be 100% reliable. There can be times that a user may want to reach something on the Internet that happens to be unavailable at the time. This problem can be on the user's end or on the server's end. We have created a simple application (app) that can be used to tell the user the status of a particular web service, indicating whether it is down or not. In this paper, we have explained how our app works and how it can be used. We have also compared our software with similar trackers to know more about the statistics behind our application.

**Keywords:** Networks, Pinging, IPs, Connectivity, Python

## 1 Introduction

### 1.1 What is Pinging?

The ping command is used to check connectivity with an IP address. More specifically, if you can reach that IP/host. When you ping something that is reachable you will get a response. Using Terminal or Command prompt, you can get statistics of your ping tests. Ping is a command or a program that is used by system administrators to check if a specific IP address is up and running and in a state to receive requests. Underneath Ping an Internet Control Message Protocol (ICMP) is used to send an Echo Request to a particular server on the Internet and looks for a response. The other uses of Ping are that it could be used for connection troubleshooting, and to find the time an echo request take to get back.

### 1.2 What is Python/ Why did we choose Python?



Figure 1: Python Logo [1]

Python is a programming language that emphasizes code readability and simplicity. Python was created by Guido van Rossum and was first implemented in 1989 [2]. Python is dynamic and allows programmers to create programs quickly and efficiently. Python has now become one of the most popular dynamic programming languages. It also supports networking features that are the main requirements for this work [2].

Python was an easy choice for us as a medium to work in, not only due to our experience working with language, but its overall ease of use. Trying to develop a networking application was simple enough in the end, due to the capability of writing directly into the computer's command line or terminal (Windows vs. OS X/Linux). This allowed us the flexibility to only have to write the ping command into the command prompt and then the command would be run as it normally would on any given computer.

### 1.3 Similar Trackers

There are other services available on the Internet that allow users to track downtime. The one that first comes to mind for us is [downdetector.com](http://downdetector.com). [Downdetector.com](http://downdetector.com) is a very useful tool that we use quite frequently. A user can simply search for something like, "Is [insert service name] down?" and the first thing they will usually see is a link to [downdetector.com](http://downdetector.com). [Downdetector.com](http://downdetector.com) collects user's status reports for a service and analyzes the data it is given to detect outages.

Another service we found on the Internet is, "Is It Down Right Now?". "Is It Down Right Now?" is used to check websites and see if they are down by pinging them. Their website will ping a website upon a user's request. [3]

There are other smaller websites on the Internet that do the same thing. Unlike [Downdetector.com](http://downdetector.com) and [Is It Down Right Now](http://isitdownrightnow.com), these other websites do not display the most popular searches and only ping what the user asks for. They may or may not give statistics of the pings. [4]

Sometimes, the services that we are looking for have their own status page. While this is usually an accurate status report from the company itself, if the website that the status page is on is down, that would be an issue. For instance, Sony provides this service for all of their PlayStation network

services. [5]

## 2 The application

We created an offline application using the Python programming language to facilitate the pinging of websites to see if communication is available. Inspiration was taken from a number of online services that exist to help in much the same way. After pinging a host, if a response is received the program will report that website is up and available, and if no response is received (after a period of time) then the host will be reported as down.

Websites/Hosts are added by the user (so that it is customizable to the user's wants/desires) and those websites/hosts are added to an array of program to iterate over at a later time. Once the user is done, the program will begin tracking, and looping over the program at a set interval. By running the ping command, we are able to see if your computer can communicate with a given web service. For the sake of readability we suppress the output of the function so that the command prompt is not filled to the brim with the results of the ping command. However, we give up showing the users any statistics of that ping.

The general results of the ping are stored in another array that are then printed out. Then, the program begins at the top of the loop again, and clears the command prompt, to again, maintain readability.

## 3 Results and Analysis

The developed program runs as intended. It takes in multiple user inputs that can be either website URLs or IP addresses. Upon inputting 'q', the tracking (pinging) begins and the program will check the given addresses and see whether they are up or down at that given time. The user can then press any key on their keyboard to stop tracking.

This program has been tested with various inputs and has proven to work consistently, as long as the address given actually exists. The program cannot distinguish between a fake address and a real server that is actually down. Both cases will just output that whatever is at the address (if anything) is down.

It is important to know that this program does not give you the statistics of the ping test like the ping command normally does. Instead, it is focused on getting a response from what it is pinging and showing that something is reachable. The program also waits a few seconds when pinging to see if the response is just coming in late before saying that what is being pinged is down. If it does all that, it is working as intended.

## 4 Challenges encountered

Due to the simplicity and ease of use the Python aims to provide, we did not run into any issues when developing our project. The extensive online community is also helpful in overcoming obstacles or any unfamiliarity with the programming language.

Having concurrent process run at the same time was an issue that we had initially run into. How can you continually loop through a program while also waiting on a user to provide input? We found out that this can be solved (like in many languages) using threads. However, while opening a separate thread could achieve our desired effect, we found an alternative method that was simpler to implement.

We had also found that implementing support for differing operating systems was more complicated than we had first anticipated. In the end we had settled to only support windows platforms. This allowed us the use of the msvcrt module built into python. This module allowed us to implement a method to detect user input but not have to stop our main loop from functioning. This makes for a much better end user experience.

There is a subtle differences in the way that operating systems use their terminal/command line to ping a host. When using python to ping a website on a windows machine, the line needs to be set up to use "-n" as part of the command, where with OS X (for apple computers) and Linux use "-c" as part of the command. To support all platforms the operating system needed to be detected at run time, to use different statements.

Both automatic and User-Reported downtime tracking programs have their pros and cons. A user-reported system such as downdetector.com works really well if the user base is large and if the service being tracked is also has a large user base. If either of those groups are too small, information provided can be unreliable, because it could be an individual's slow internet connected or a mistake on their end.

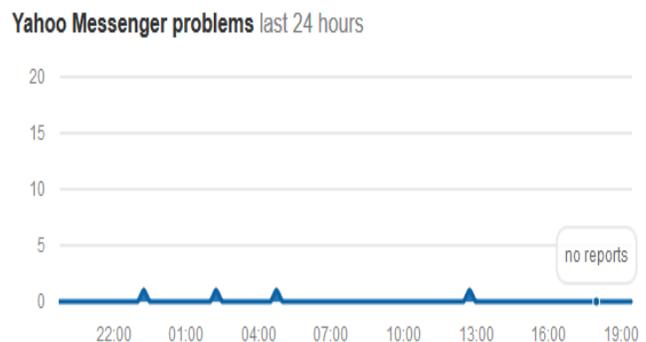


Figure 2: Small population reports [3]

The same is represented in figure 2. The figure 2 above can be interpreted in two ways: either Yahoo Messenger is perfect and no one is really having any issues with it, or the user base for the software is so small that no one is coming to this site to report problems with it. With no real way to differentiate between the two, it is hard to come to any solid conclusions. Unlike, the figure 3 below where a lot of user reports are provided, and the likelihood of there actually being issues is more dependable.

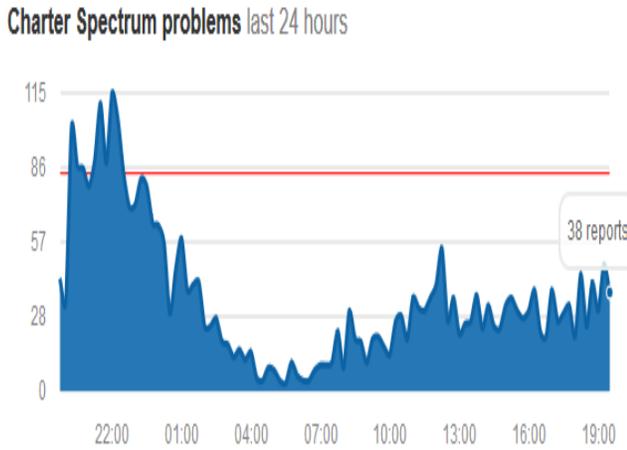


Figure 3: Larger population reports [3]

Web services that ping websites (much like our own product) such as isitdownrightnow.com can provide the kind of information that user-reported sites such as downdetector.com do. However, they do not require a large user base for reliable data. They ping a website upon request, and then you can find out if the website/host is down or not. They are not all created equally. However, for instance, isitdownright.com is restricted to only pinging websites and cannot ping specific IPs. Finding out if a game server is down cannot be done, the system is reliant on using host names.



Figure 4: Example of a website that is up on isitdownrightnow.com [4]



Figure 5: Example of a website that is up on isitdownrightnow.com [4]

Our software poses a unique advantage over a service such as isitdownrightnow.com. Due to the fact that their service is hosted on the web, they can also be susceptible to being down or having long delays in connecting. Our application works on the machine itself and does not necessitate the use of a web browser. Our software does have some downsides but they do not prevent the software from fulfilling its purpose. Our software must be left on and continuously pinging if a user wants to ping something throughout the day. This is not really something we would expect users to do because our software is more focused on the current status rather than a log of statuses. The software also does not give statistics with the pings but we never intended for it to do that. The software also only works on Windows so someone would not be able to use it on OS X or on mobile devices.

## 5 Conclusion

This paper presents an application to help users track whether or not the website or host of their choice is available. The application is giving good results, and when compared to other services or applications, it has its own set of pros and cons. Our application works on the machine itself and does not necessitate the use of a web browser.

## 6 References

- [1] "Welcome to Python.org." *Python.org*. Python, 2001. Web. 15 Apr. 2017. <<https://www.python.org/>>.
- [2] Rossum, Guido Van. "A Brief Timeline of Python." *The History of Python*. N.p., 01 Jan. 2009. Web. 17 Apr. 2017. <<http://python-history.blogspot.com/2009/01/brief-timeline-of-python.html>>.
- [3] "Downdetector.com." *Downdetector.com*. Serinus42, n.d. Web. 2 Apr. 2017. <<http://downdetector.com/about-us/>>.
- [4] "Is It Down Right Now? Website Down or Not?" *Is It Down Right Now*. N.p., 2012. Web. 15 Apr. 2017. <<http://www.isitdownrightnow.com/>>.
- [5] "NetworkService Status." *PlayStation Network Status*. Sony, n.d. Web. 17 Apr. 2017. <<https://status.playstation.com/en-US/>>.
- [6] Evanoff Lauren, Nicole Hatch, Gagneja K.K., "Home Network Security: Beginner vs Advanced", ICWN, Las Vegas, USA, July 27-30, 2015.
- [7] Kanwal G, "Pairwise Key Distribution Scheme for Two-Tier Sensor Networks", IEEE ICNC, Honolulu, Hawaii, USA, pages 1081-1086, Feb. 3-6, 2014.
- [8] Nygard K., Gagneja K., "Energy Efficient Approach with Integrated Key Management Scheme for Wireless Sensor Networks", ACM MOBIHOC, Bangalore, India, pages 13-18, July 29, 2013.
- [9] Gagneja K.K. and Nygard K., "Heuristic Clustering with Secured Routing in Heterogeneous Sensor Networks", IEEE SECON, New Orleans, USA, pages 9-16, June 24-26, 2013.
- [10] Gagneja K.K., "Secure Communication Scheme for Wireless Sensor Networks to maintain Anonymity", IEEE ICNC, Anaheim, California, USA, Feb. 16-19, 2015.
- [11] Gagneja K.K. and Nygard K., "A QoS based Heuristics for Clustering in Two-Tier Sensor Networks", IEEE FedCSIS 2012, Wroclaw, Poland, pages 779-784, Sept. 9-12, 2012.
- [12] Gagneja K.K. and Nygard K., "Tabu-Voronoi Clustering Heuristics with Key Management Scheme for Heterogeneous Sensor Networks", IEEE ICUFN 2012, Phuket, Thailand, pages 46-51, July 4-6, 2012.
- [13] Gagneja K.K., "Knowing the Ransomware and Building Defense Against it - Specific to HealthCare Institutes", IEEE MobiSecServ, Miami, USA, Feb. 11-12, 2017.
- [14] Runia Max, Gagneja K.K., "Raspberry Pi Webserver", ESA, Las Vegas, USA, July 27-30, 2015.
- [15] Singh Arvinderpal, Gagneja K. K., "Incident Response through Behavioral Science: An Industrial approach", IEEE CSCSI, Las Vegas, USA, Dec. 7-9, 2015.
- [16] Gagneja K.K. and Nygard K., "Pairwise Post Deployment Key Management Scheme for Heterogeneous Sensor Networks", 13th IEEE WoWMoM 2012, San Francisco, California, USA, pages 1-2, June 25-28, 2012.
- [17] Gagneja K.K. and Nygard K., "Key Management Scheme for Routing in Clustered Heterogeneous Sensor Networks", IEEE NTMS 2012, Security Track, Istanbul, Turkey, pp. 1-5, 7-10 May, 2012.
- [18] Gagneja K.K., "Global Perspective of Security Breaches in Facebook", FECS, Las Vegas, USA, July 21-24, 2014.
- [19] Inter-Process Communication with Sockets. (n.d.). Retrieved April 17, 2017, from [http://www.it.uom.gr/teaching/distributedSite/dsIdaLiu/labs/lab2\\_1/sockets.html](http://www.it.uom.gr/teaching/distributedSite/dsIdaLiu/labs/lab2_1/sockets.html)
- [20] Python Networking Programming. Retrieved April 18, 2017, from [https://www.tutorialspoint.com/python/python\\_networking.htm](https://www.tutorialspoint.com/python/python_networking.htm)
- [21] Network socket. (2017, April 13). Retrieved April 18, 2017, from [https://en.wikipedia.org/wiki/Network\\_socket#History](https://en.wikipedia.org/wiki/Network_socket#History)

## 7 Code Appendix

The following code chunks are the main methods that give functionality to our code as well as the main loop of our code that allow the system to ping websites continually.

`cls()`: Allows the system to clear the command prompt (note: does not work in the Python editing environment.)

`printReslts()`: Prints the results of the array that was added to via the `pingTest()` method.

`pingTest()`: Loops over each address in the array that was added to via the user's input. It makes sure that the information that is printed is suppressed to just the status of the ping.

```

#-----#
#FUNCTIONS#
#-----#
def cls():
    os.system('cls' if os.name=='nt' else 'clear')

def printResults():
    for x in range (0, len(resultArray)):
        print(resultArray[x])

def pingTest():

    for x in range(0,len(addressArray)):
        proc = subprocess.Popen(
            ['ping', '-n', '1', addressArray[x]],
            stdout=subprocess.PIPE)
        stdout, stderr = proc.communicate()
        if proc.returncode == 0:
            print('{} is UP'.format(addressArray[x]))
        else:
            print('{} is DOWN'.format(addressArray[x]))

done = False
timepinged = 0
while not done:

    cls()
    timepinged+=1
    print("Press any key keyboard at anytime to stop tracking.")
    print(timepinged)
    pingTest()
    time.sleep(5)

    if msvcrt.kbhit():
        print ("Thank you for using the ping tracker!")
        done = True
        input ()

```

Program in use:

```
Please insert the websites you would like the program to track  
Insert 'q' at any time to begin tracking.
```

```
google.com  
bing.com  
battle.net  
youtube.com  
24.105.30.129  
211.115.104.1  
wikipedia.com  
theonion.com  
542655.2352.23552.1245  
q
```

```
Press any key keyboard at anytime to stop tracking.
```

```
1  
google.com is UP  
bing.com is UP  
battle.net is UP  
youtube.com is UP  
24.105.30.129 is UP  
211.115.104.1 is DOWN  
wikipedia.com is UP  
theonion.com is UP  
542655.2352.23552.1245 is DOWN
```

```
Press any key keyboard at anytime to stop tracking.
```

```
3  
google.com is UP  
bing.com is UP  
battle.net is UP  
youtube.com is UP  
24.105.30.129 is UP  
211.115.104.1 is DOWN  
wikipedia.com is UP  
theonion.com is UP  
542655.2352.23552.1245 is DOWN  
Thank you for using the ping tracker!  
a
```