Networking Fundamentals

CENTRE OF EXCELLENCE IN IT

©Centre for Development of Advanced Computing (C-DAC)

Network Troubleshooting

Introduction

- Many layers, devices, and protocols are involved when it comes to computer networking.

- Alot of the devices and protocols that we have gone through have built-in functionalities which help protect against some of these issues. These functionalities are known as error detection and error recovery.

Error-detection → the ability for a protocol or program to determine that something went wrong.

Error-recovery → the ability for a protocol or program to attempt to fix it.

Network Troubleshooting

- In any network, there has to be communication. No communication means that the network is down.

- The inability to establish a connection has us tracing, analyzing, and solving to come up with a solution.

This is known as network troubleshooting.

Network Troubleshooting

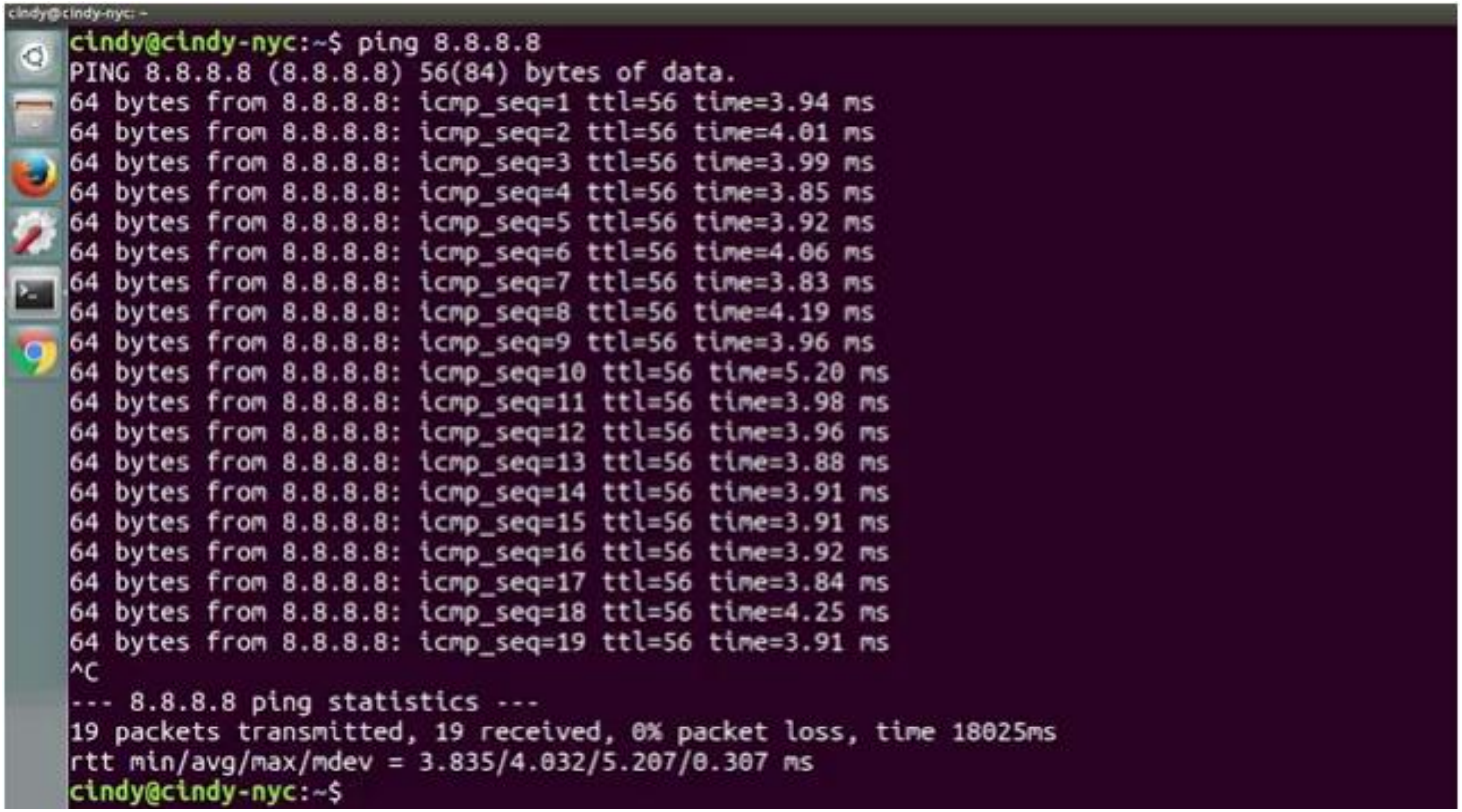
Verifying Connectivity

Troubleshooting a network

1. Check the hardware
2. Use ipconfig
3. Use ping and traceroute
4. Use Netcat and Test-NetConnection to test port connectivity.
5. Use nslookup to perform DNS check

Network Troubleshooting

Ping: Internet Control Message Protocol - Linux

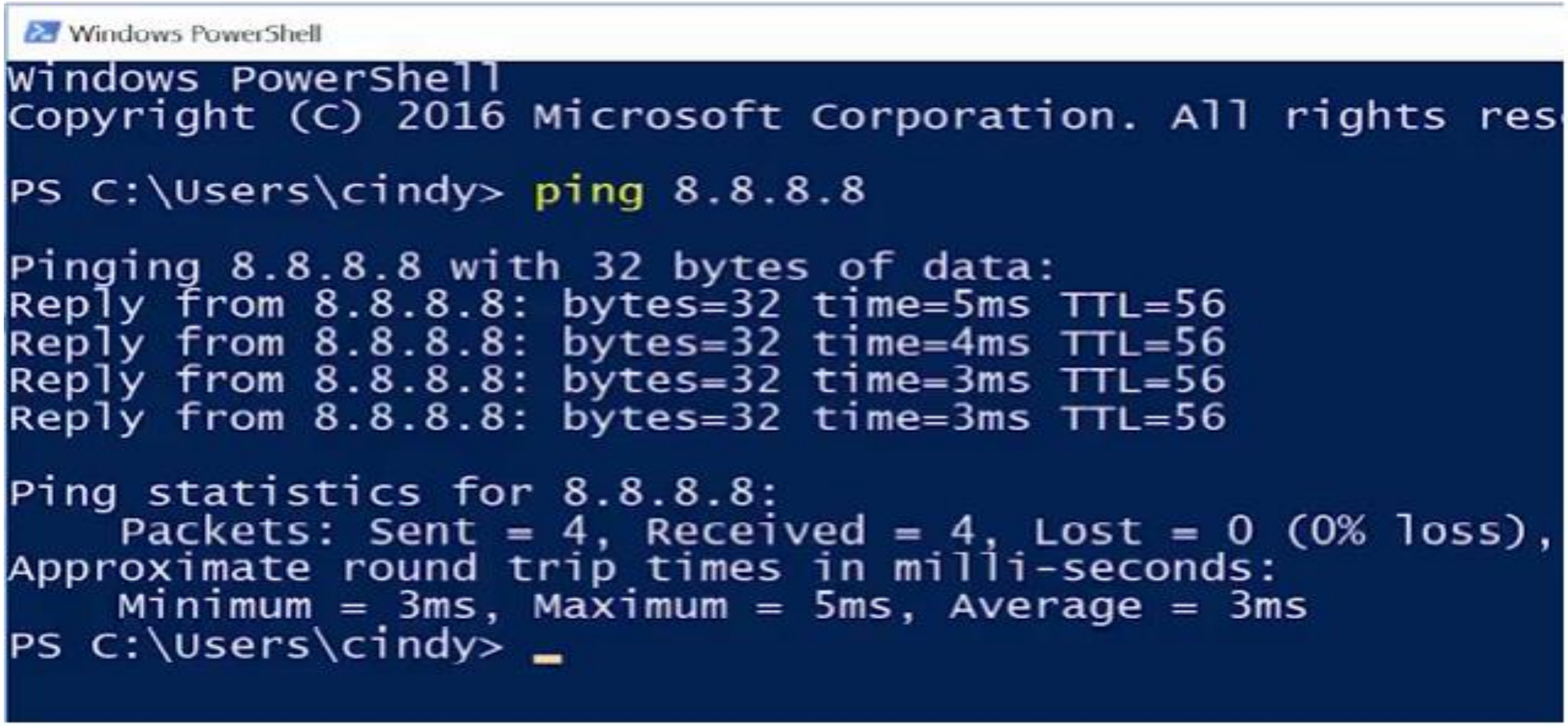


```
cindy@cindy-nyc:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=56 time=3.94 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=56 time=4.01 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=56 time=3.99 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=56 time=3.85 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=56 time=3.92 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=56 time=4.06 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=56 time=3.83 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=56 time=4.19 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=56 time=3.96 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=56 time=5.20 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=56 time=3.98 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=56 time=3.96 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=56 time=3.88 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=56 time=3.91 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=56 time=3.91 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=56 time=3.92 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=56 time=3.84 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=56 time=4.25 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=56 time=3.91 ms
^C
--- 8.8.8.8 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18025ms
rtt min/avg/max/mdev = 3.835/4.032/5.207/0.307 ms
cindy@cindy-nyc:~$
```

©Centre for Development of Advanced Computing (C-DAC)

Network Troubleshooting

Ping: Internet Control Message Protocol – Windows

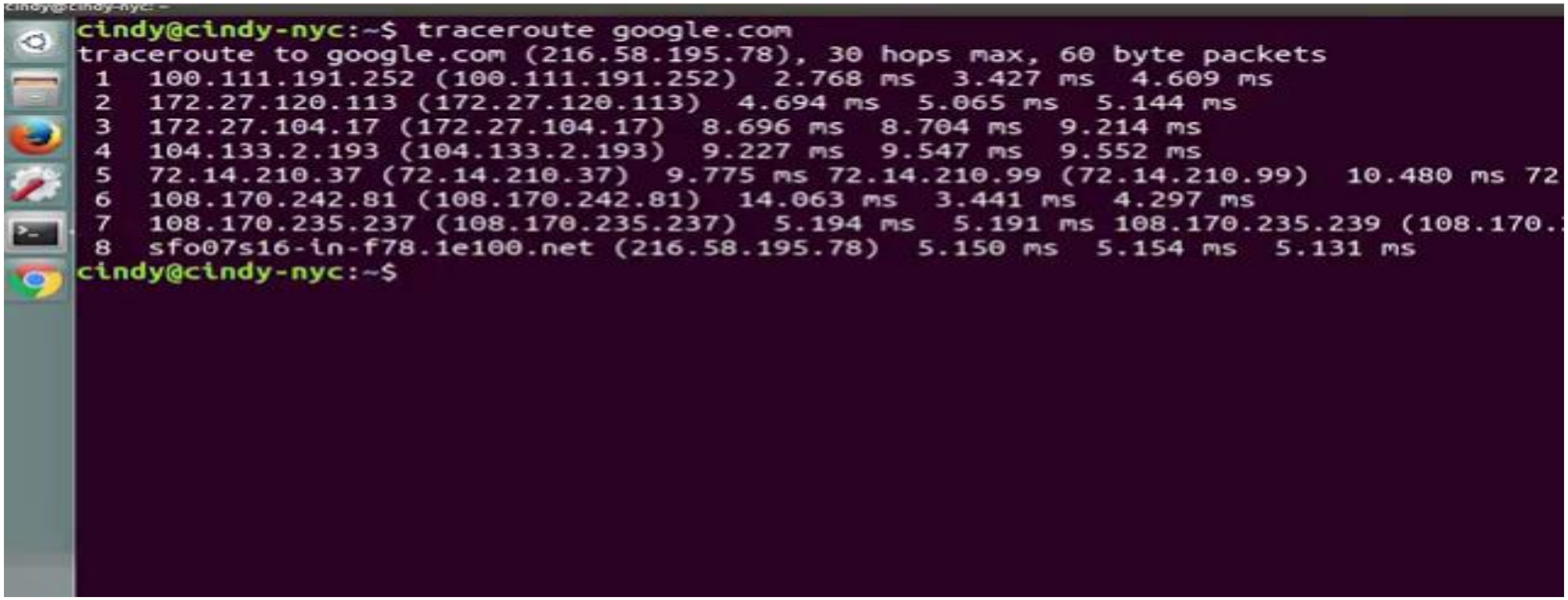


```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights res

PS C:\Users\cindy> ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=5ms TTL=56
Reply from 8.8.8.8: bytes=32 time=4ms TTL=56
Reply from 8.8.8.8: bytes=32 time=3ms TTL=56
Reply from 8.8.8.8: bytes=32 time=3ms TTL=56

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 5ms, Average = 3ms
PS C:\Users\cindy> _
```

Network Troubleshooting

Traceroute

On Linux and MacOS, traceroute sends UDP packets to very high port numbers.

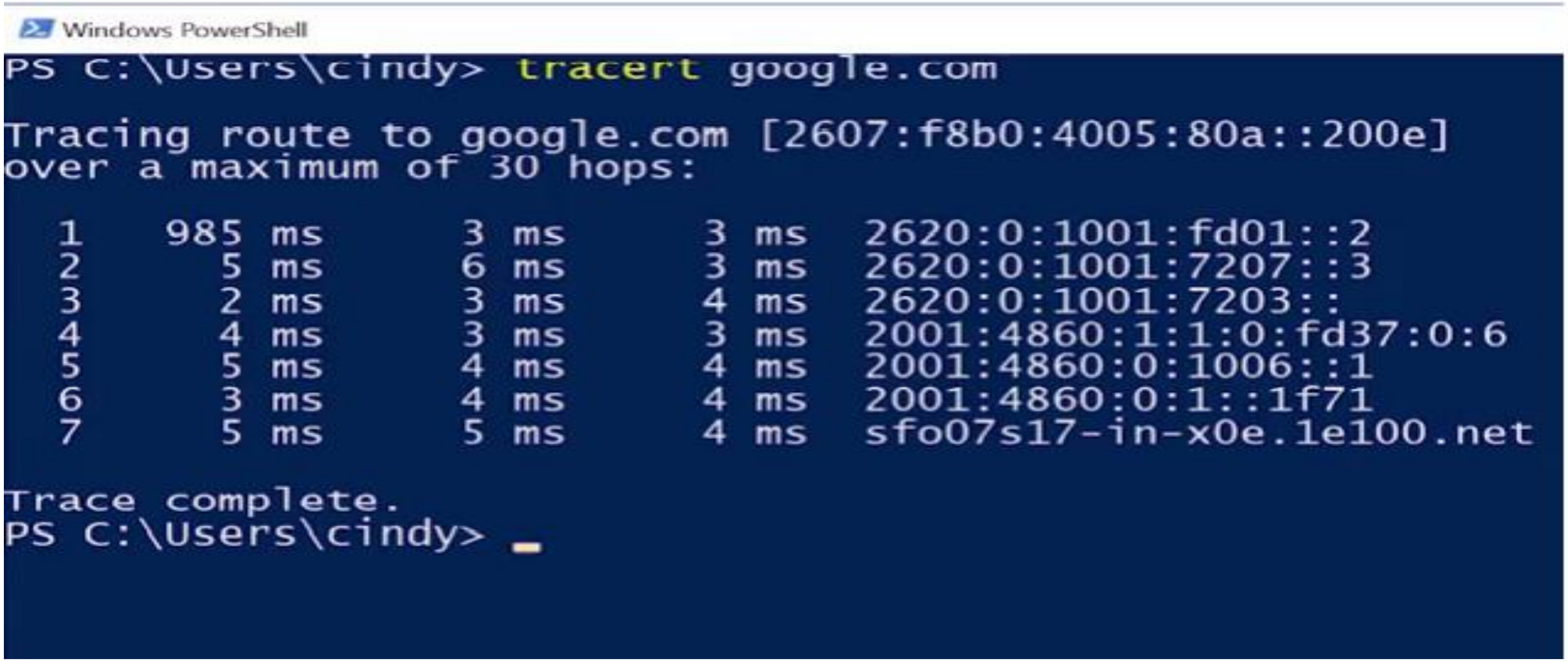


```
cindy@cindy-nyc:~$ traceroute google.com
traceroute to google.com (216.58.195.78), 30 hops max, 60 byte packets
 1  100.111.191.252 (100.111.191.252)  2.768 ms  3.427 ms  4.609 ms
 2  172.27.120.113 (172.27.120.113)  4.694 ms  5.065 ms  5.144 ms
 3  172.27.104.17 (172.27.104.17)  8.696 ms  8.704 ms  9.214 ms
 4  104.133.2.193 (104.133.2.193)  9.227 ms  9.547 ms  9.552 ms
 5  72.14.210.37 (72.14.210.37)  9.775 ms 72.14.210.99 (72.14.210.99)  10.480 ms 72
 6  108.170.242.81 (108.170.242.81)  14.063 ms  3.441 ms  4.297 ms
 7  108.170.235.237 (108.170.235.237)  5.194 ms  5.191 ms 108.170.235.239 (108.170.
 8  sfo07s16-in-f78.1e100.net (216.58.195.78)  5.150 ms  5.154 ms  5.131 ms
cindy@cindy-nyc:~$
```

Network Troubleshooting

Traceroute

On Windows, the command has a shortened name **tracert**, and defaults to using ICMP echo request.

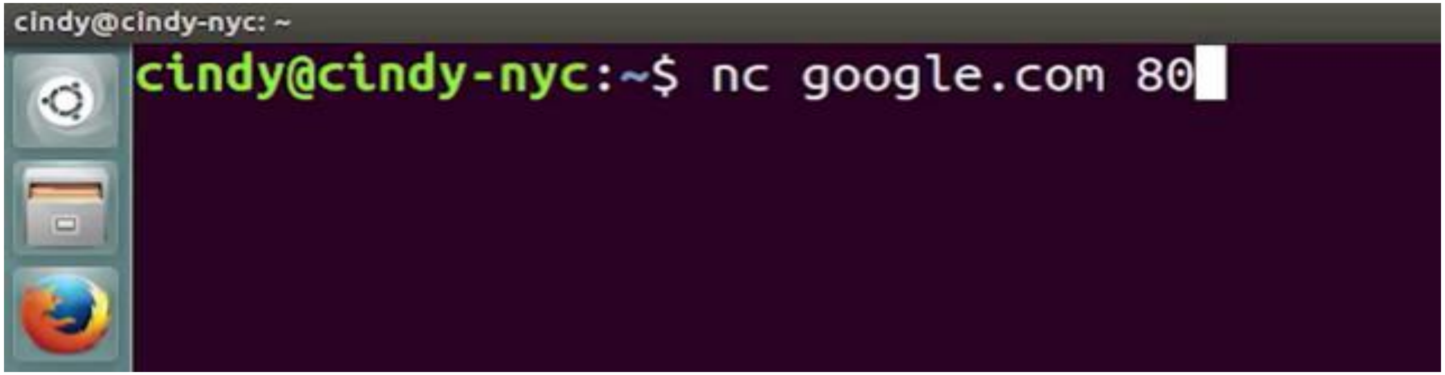


```
PS C:\Users\cindy> tracert google.com

Tracing route to google.com [2607:f8b0:4005:80a::200e]
over a maximum of 30 hops:

  1   985 ms     3 ms     3 ms  2620:0:1001:fd01::2
  2     5 ms     6 ms     3 ms  2620:0:1001:7207::3
  3     2 ms     3 ms     4 ms  2620:0:1001:7203::
  4     4 ms     3 ms     3 ms  2001:4860:1:1:0:fd37:0:6
  5     5 ms     4 ms     4 ms  2001:4860:0:1006::1
  6     3 ms     4 ms     4 ms  2001:4860:0:1::1f71
  7     5 ms     5 ms     4 ms  sfo07s17-in-x0e.1e100.net

Trace complete.
PS C:\Users\cindy>
```

Network Troubleshooting

Testing Port Connectivity

Sometimes, you need to know if things are working at the transport layer. For this, there are two powerful tools at your disposal. Netcat on Linux and Mac OS and Test-NetConnection on Windows. The Netcat tool can be run through the command nc, and has two mandatory arguments, a host and a port.

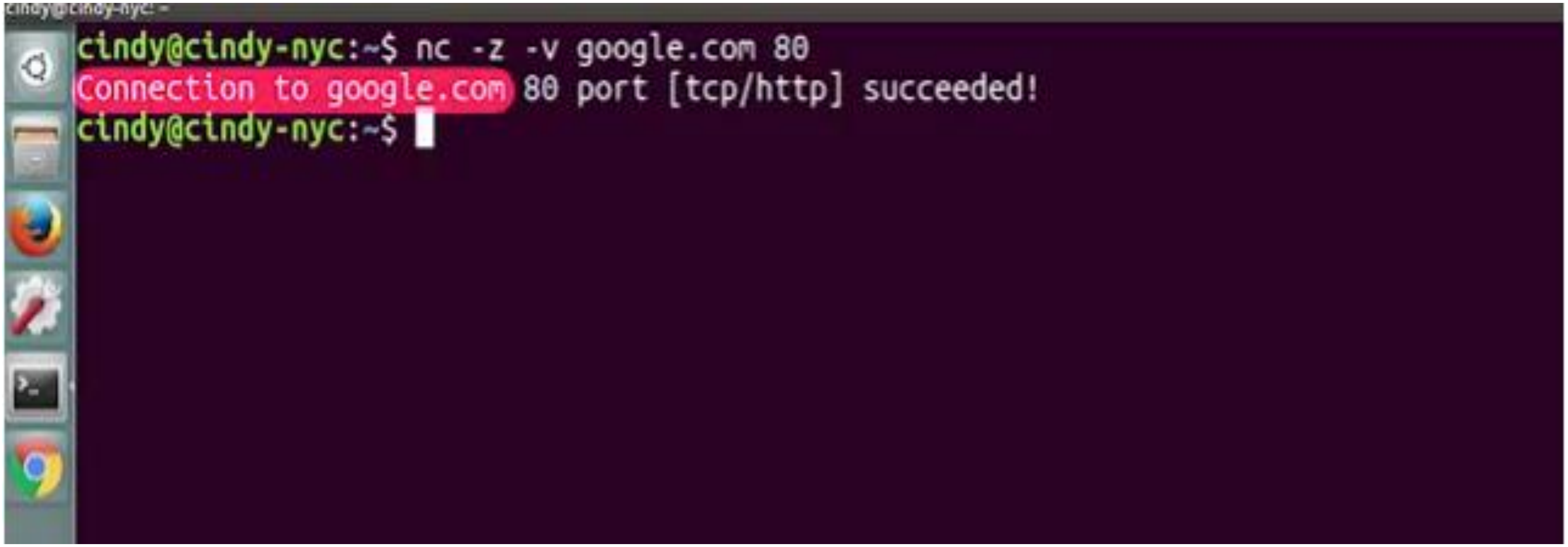


```
cindy@cindy-nyc:~$ nc google.com 80
```

Network Troubleshooting

Testing Port Connectivity

So by issuing the Netcat command with the -Z and -V flags, the command's output will simply tell you if a connection to the port in question is possible or not.



```
cindy@cindy-nyc:~$ nc -z -v google.com 80
Connection to google.com 80 port [tcp/http] succeeded!
cindy@cindy-nyc:~$
```

©Centre for Development of Advanced Computing (C-DAC)

Network Troubleshooting

Testing Port Connectivity

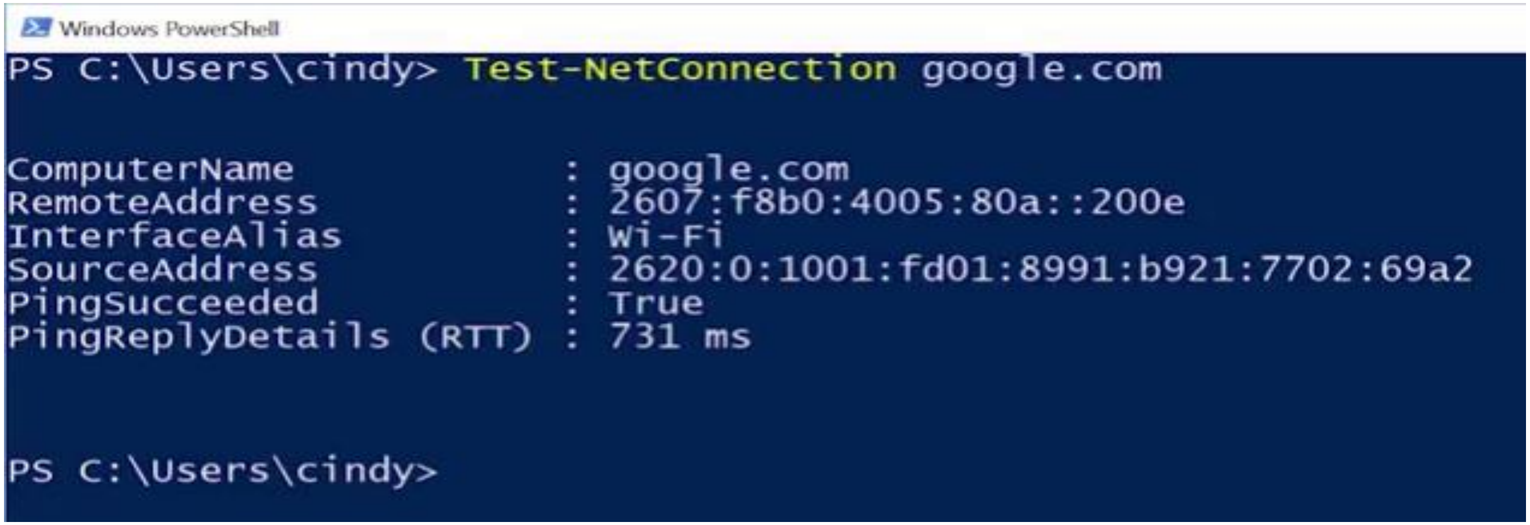
- On Windows, Test-NetConnection is a command with some of the similar functionality.

- Running Test-NetConnection with only a host specified will default to using an ICMP echo request, much like the program ping. But, it will display way more data, including the data link layer protocol being used.

- Issuing Test-NetConnection with the -port flag, you can ask it to test connectivity to a specific port.

Network Troubleshooting

Testing Port Connectivity - Windows



Windows PowerShell

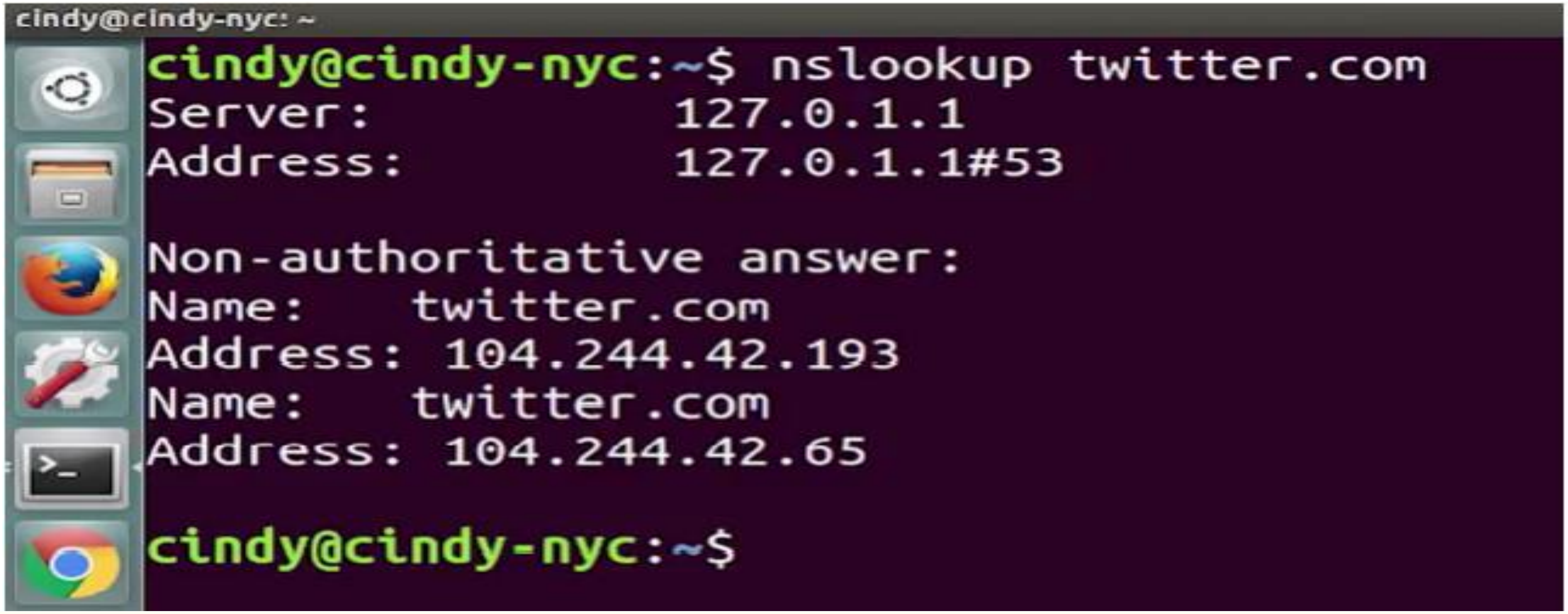
```
PS C:\Users\cindy> Test-NetConnection google.com


ComputerName           : google.com
RemoteAddress          : 2607:f8b0:4005:80a::200e
InterfaceAlias         : Wi-Fi
SourceAddress          : 2620:0:1001:fd01:8991:b921:7702:69a2
PingSucceeded          : True
PingReplyDetails (RTT) : 731 ms



PS C:\Users\cindy>
```

It’s important to call out that both Netcat and Test-NetConnection are way more powerful than the brief port connectivity examples we’ve covered.

Network Troubleshooting

Name Resolution Tools

Let's say you needed to know the IP address for a twitter.com. You would just enter nslookup twitter.com and the record would be returned.

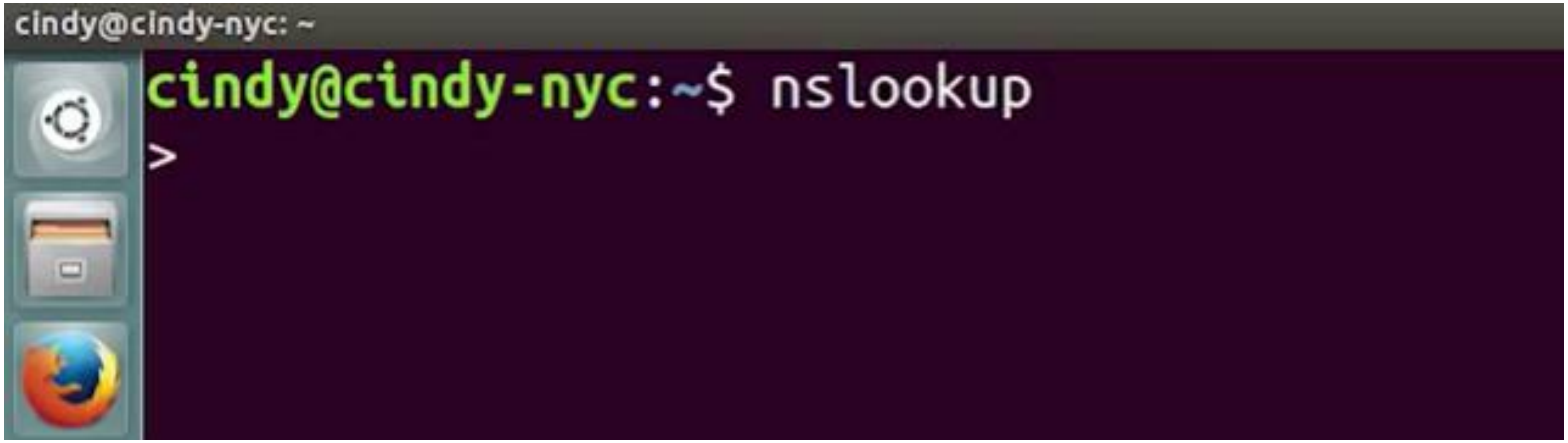


```
cindy@cindy-nyc:~$ nslookup twitter.com
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
Name:   twitter.com
Address: 104.244.42.193
Name:   twitter.com
Address: 104.244.42.65

cindy@cindy-nyc:~$
```

Network Troubleshooting

Name Resolution Tools

Nslookup is way more powerful than just that. It includes an interactive mode that lets you set additional options and run lots of queries/requests in a row. To start an interactive nslookup session, you just enter nslookup, without any hostname following it. You should see an angle bracket acting as your prompt.



```
cindy@cindy-nyc:~$ nslookup
>
```

Network Troubleshooting

Task:

Go through the mentioned tools/commands and practice on both Windows and Linux Systems.