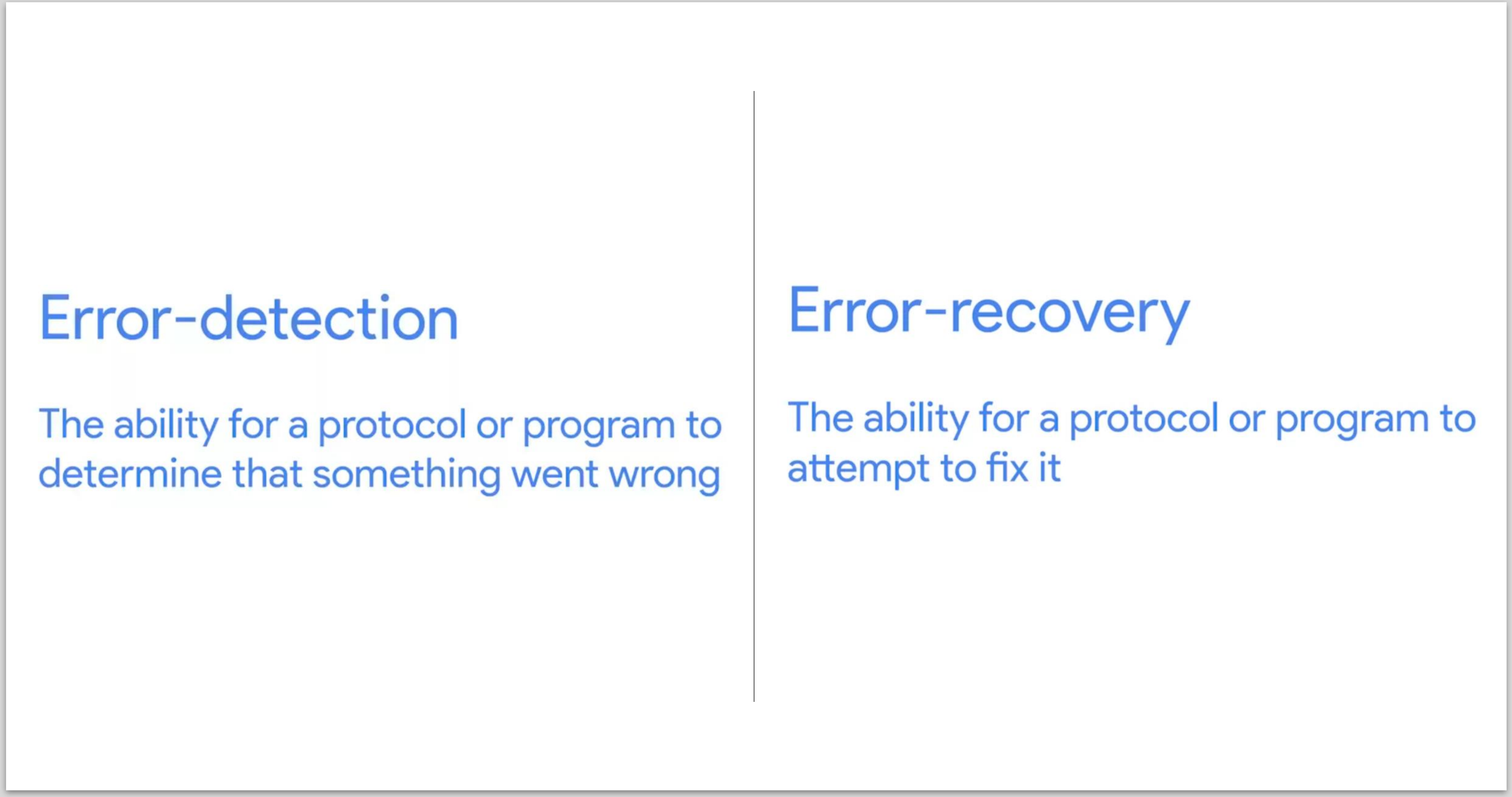


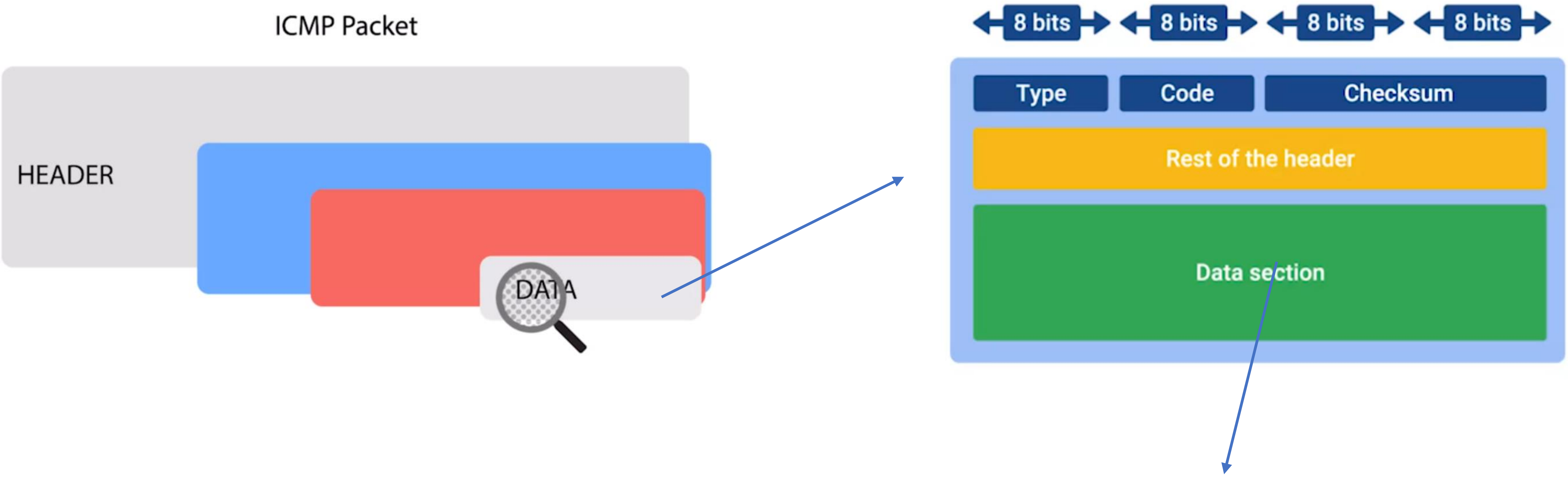
## Error-detection

The ability for a protocol or program to determine that something went wrong

## Error-recovery

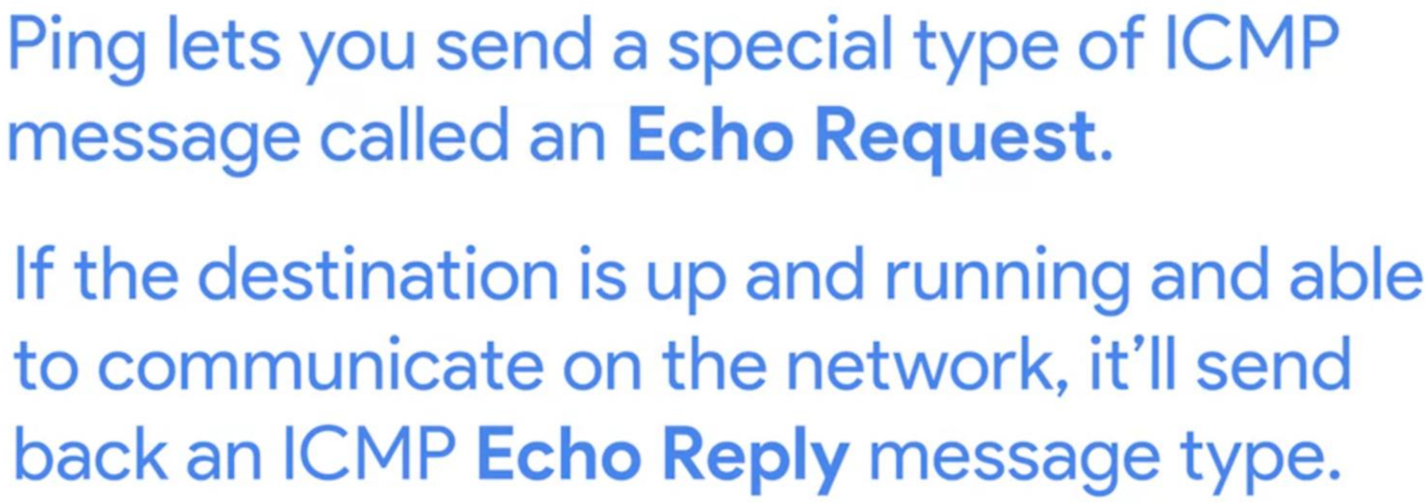
The ability for a protocol or program to attempt to fix it

# ICMP(Internet Control Message Protocol)



The payload for an ICMP packet exists entirely so that the recipient of the message knows which of their transmissions caused the error being reported.

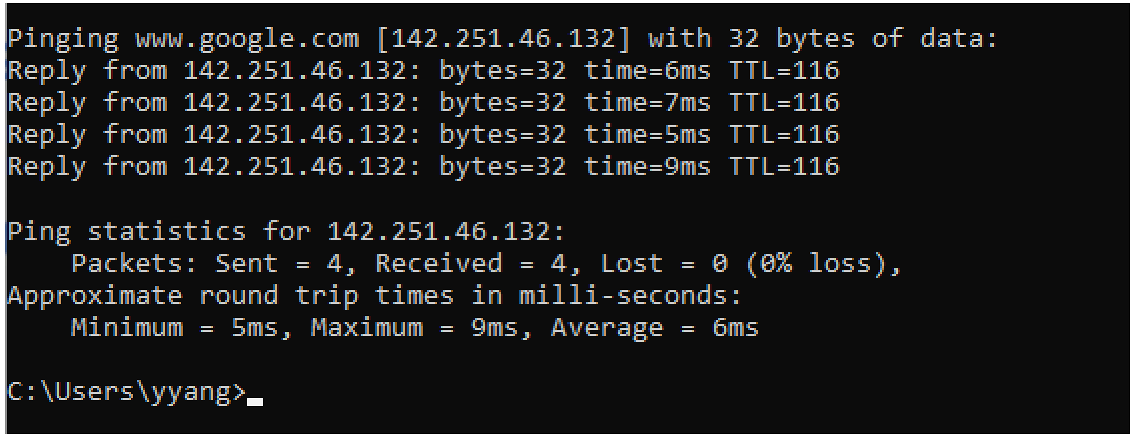
# Ping



Ping lets you send a special type of ICMP message called an **Echo Request.**

If the destination is up and running and able to communicate on the network, it'll send back an ICMP **Echo Reply** message type.

Determine whether you can reach a certain computer, and know the general quality of the connection

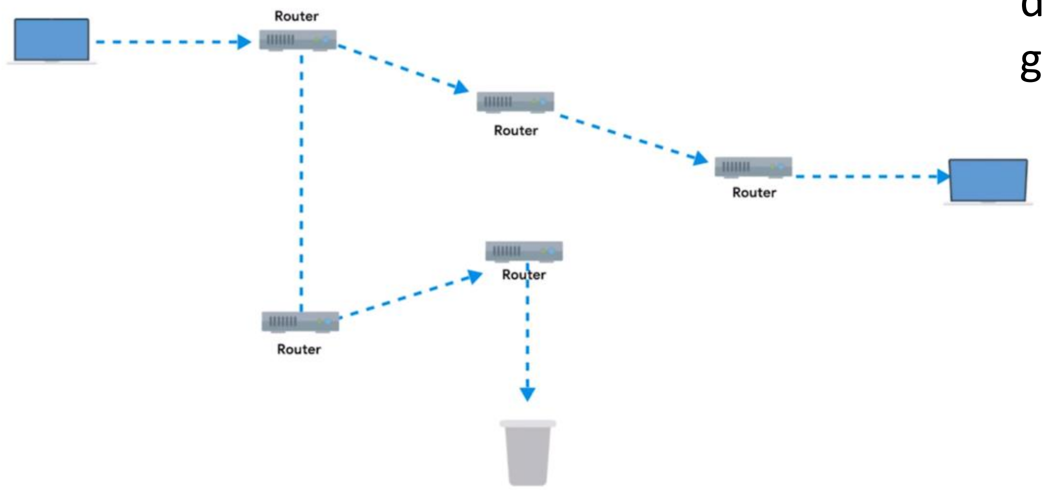


```
Pinging www.google.com [142.251.46.132] with 32 bytes of data:
Reply from 142.251.46.132: bytes=32 time=6ms TTL=116
Reply from 142.251.46.132: bytes=32 time=7ms TTL=116
Reply from 142.251.46.132: bytes=32 time=5ms TTL=116
Reply from 142.251.46.132: bytes=32 time=9ms TTL=116

Ping statistics for 142.251.46.132:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum = 9ms, Average = 6ms

C:\Users\yyang>_
```

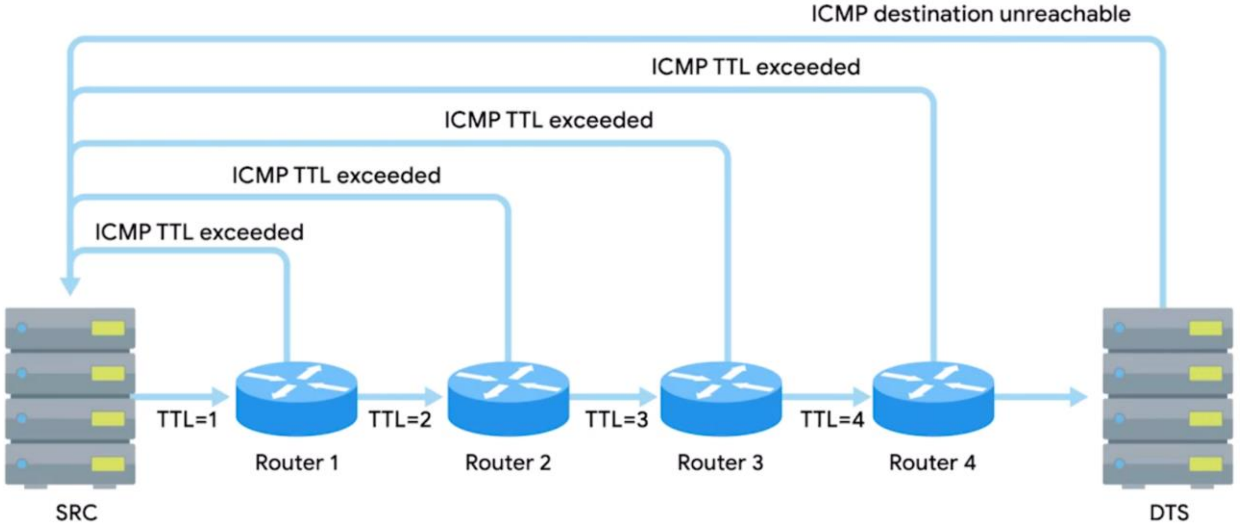
In windows, ping tool would send 4 Echo Request. In Linux, user have to interrupt it manually to make it stop.

# Traceroute

**Ping** could only tell you the overall quality of the connection but would **not let you know which node is unreachable** when disconnection happened. **Traceroute is the solution**.

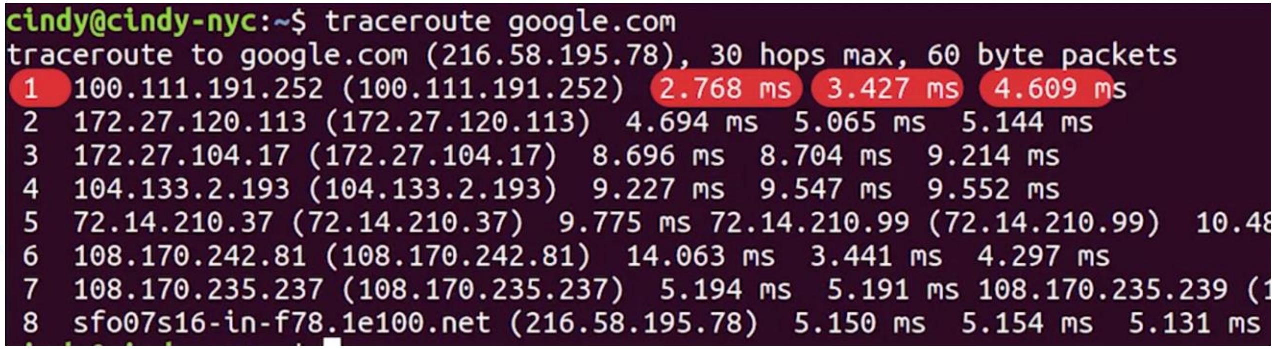


Traceroute set different TTL (when TTL reach to 0, the data would be discarded, and send back “time out” message to original node) to get the time of data delivery to each node



## Traceroute

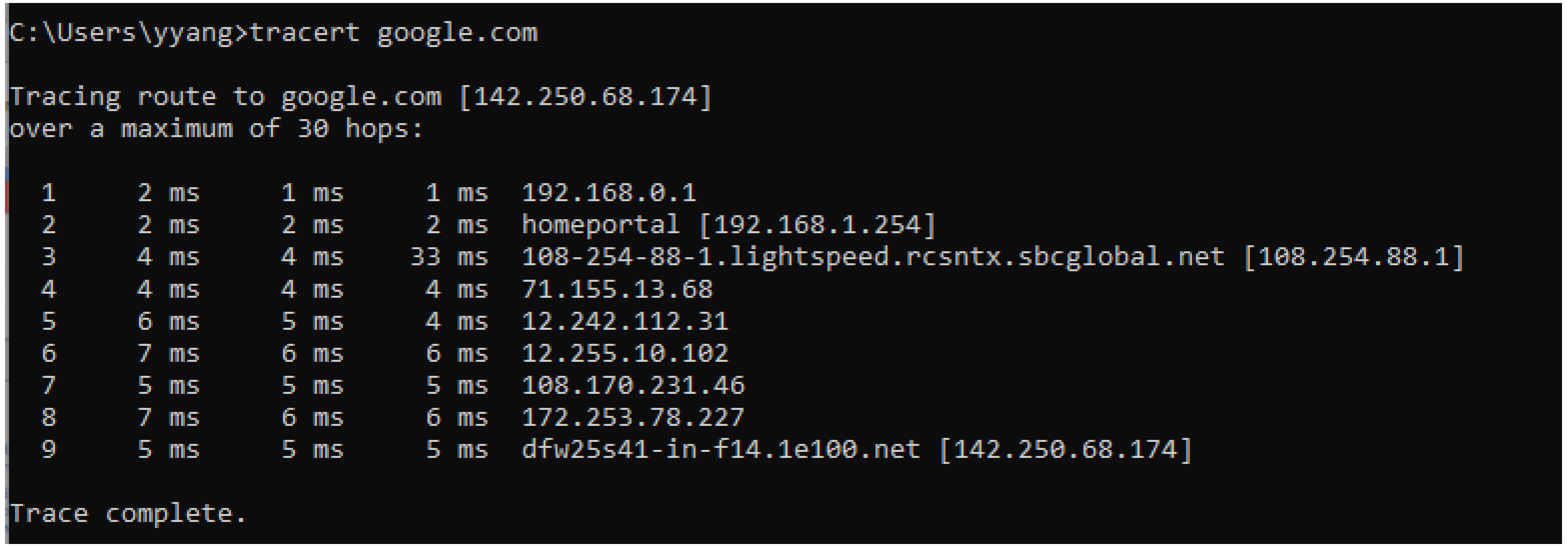
A utility that lets you discover the path between two nodes, and gives you information about each hop along the way



```
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)  10.48
 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 (1
 8  sfo07s16-in-f78.1e100.net (216.58.195.78)  5.150 ms  5.154 ms  5.131 ms
```

# Traceroute

In windows, it has a shorter name, tracert



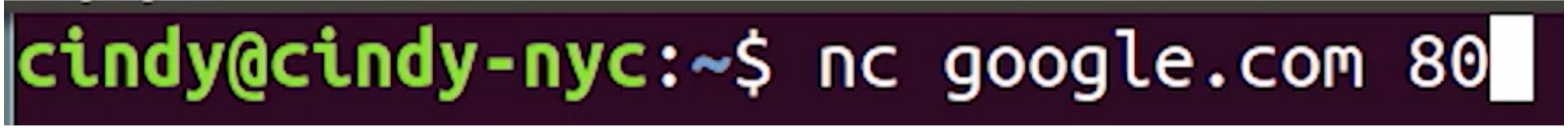
```
C:\Users\yyang>tracert google.com

Tracing route to google.com [142.250.68.174]
over a maximum of 30 hops:

  1     2 ms     1 ms     1 ms  192.168.0.1
  2     2 ms     2 ms     2 ms  homeportal [192.168.1.254]
  3     4 ms     4 ms    33 ms  108-254-88-1.lightspeed.rcsntx.sbcglobal.net [108.254.88.1]
  4     4 ms     4 ms     4 ms  71.155.13.68
  5     6 ms     5 ms     4 ms  12.242.112.31
  6     7 ms     6 ms     6 ms  12.255.10.102
  7     5 ms     5 ms     5 ms  108.170.231.46
  8     7 ms     6 ms     6 ms  172.253.78.227
  9     5 ms     5 ms     5 ms  dfw25s41-in-f14.1e100.net [142.250.68.174]

Trace complete.
```

Instead of normal traceroute, “pathping” in Windows and “mtr” in Linux and MacOS act as long running traceroutes

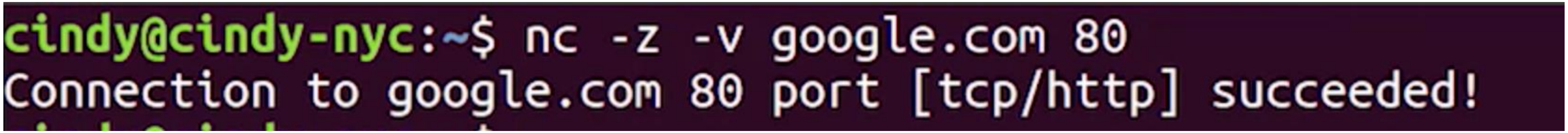
# Testing Port Connectivity

The previous methods are used for **network layer**, but sometimes the error may occur at **transport layer**.
For those problems, there are two powerful diagnose tools:

1. **Netcat** – Linux /MacOS (nc)
    1. If it works (connection established), we will see a blinking cursor waiting for more inputs, otherwise the command would exit. **(-v** for verbose, **-z** for zero input and output)

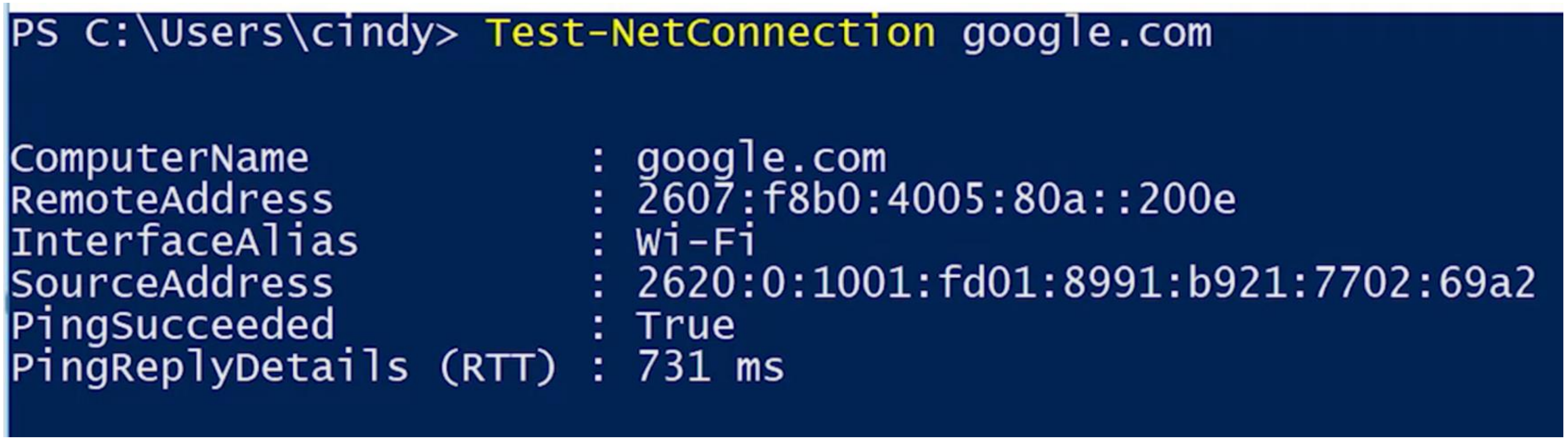


```
cindy@cindy-nyc:~$ nc google.com 80
```



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

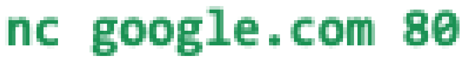
**2. Test-NetConnection** - Windows



```
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
```

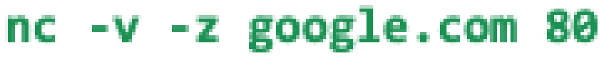
# Supplemental Reading for Testing Port Connectivity

Sometimes, you need to know if network connectivity is working at the transport layer. For this, there are two super powerful tools at your disposal: [Netcat (nc)]() on Linux and macOS, 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. Running this command would try to establish a connection on port 80 to google.com:



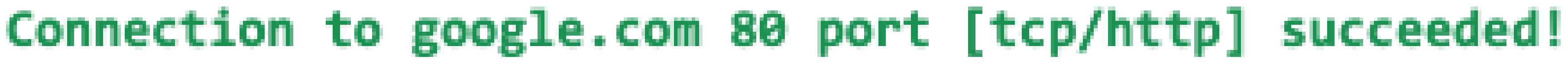
```
nc google.com 80
```

If the connection fails, the command will exit. If it succeeds, you'll see a blinking cursor, waiting for more input. This is a way for you to actually send application layer data to the listening service from your own keyboard. If you're really only curious about the status of a report, you can issue the command, with a -z flag, which stands for zero input/output mode. A -v flag, which stands for verbose, is also useful in this scenario. So now, the command looks like this:



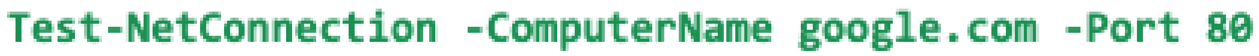
```
nc -v -z google.com 80
```

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, like this:



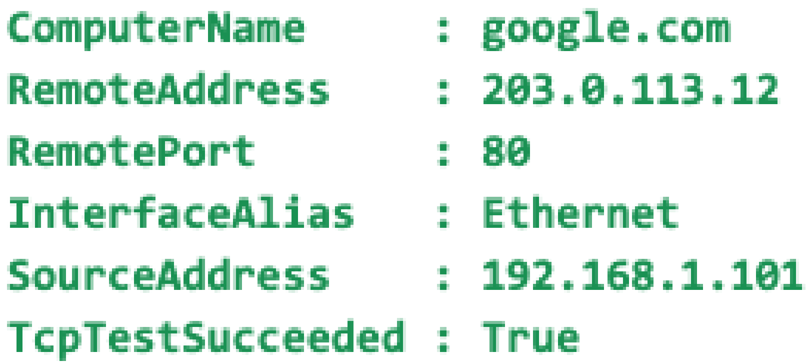
```
Connection to google.com 80 port [tcp/http] succeeded!
```

On Windows, Test-NetConnection is a command with some similar functionality. If you run Test-NetConnection with only a host specified, it 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. When you issue Test-NetConnection with the -Port flag, you can ask it to test connectivity to a specific port. For example, this command tests a TCP connection to google.com:



```
Test-NetConnection -ComputerName google.com -Port 80
```

Test-NetConnection will return output that looks something like this:

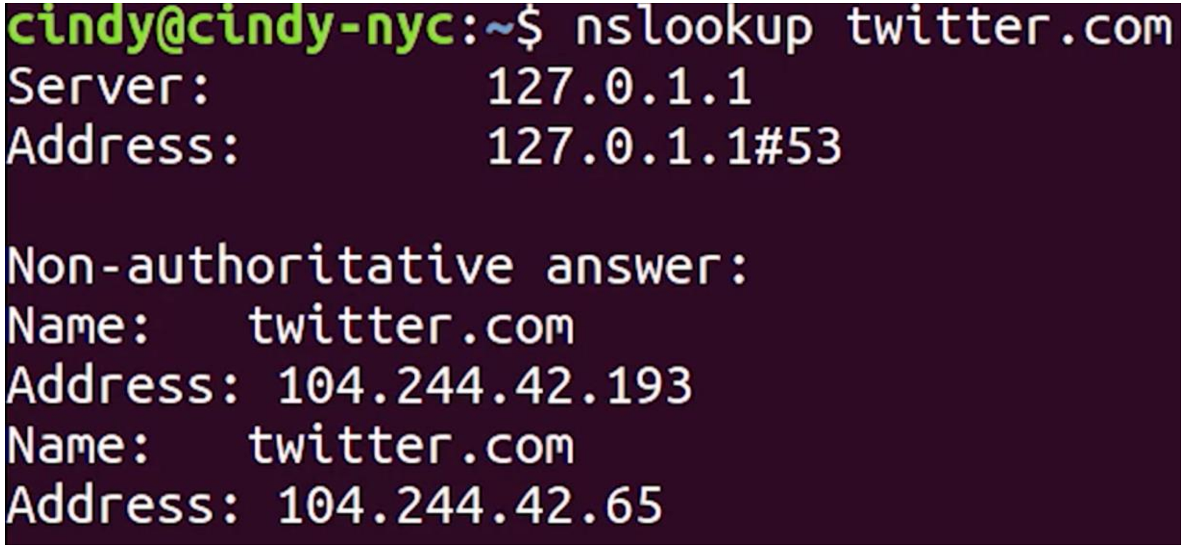


```
ComputerName     : google.com
RemoteAddress    : 203.0.113.12
RemotePort       : 80
InterfaceAlias   : Ethernet
SourceAddress    : 192.168.1.101
TcpTestSucceeded : True
```

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 here. In fact, they're such complex tools that covering all of their functionality would be too much for one video. You should read up about all of the other things these super powerful tools can do in the Wikipedia article for Netcat (nc), and in the documentation for Test-NetConnection.

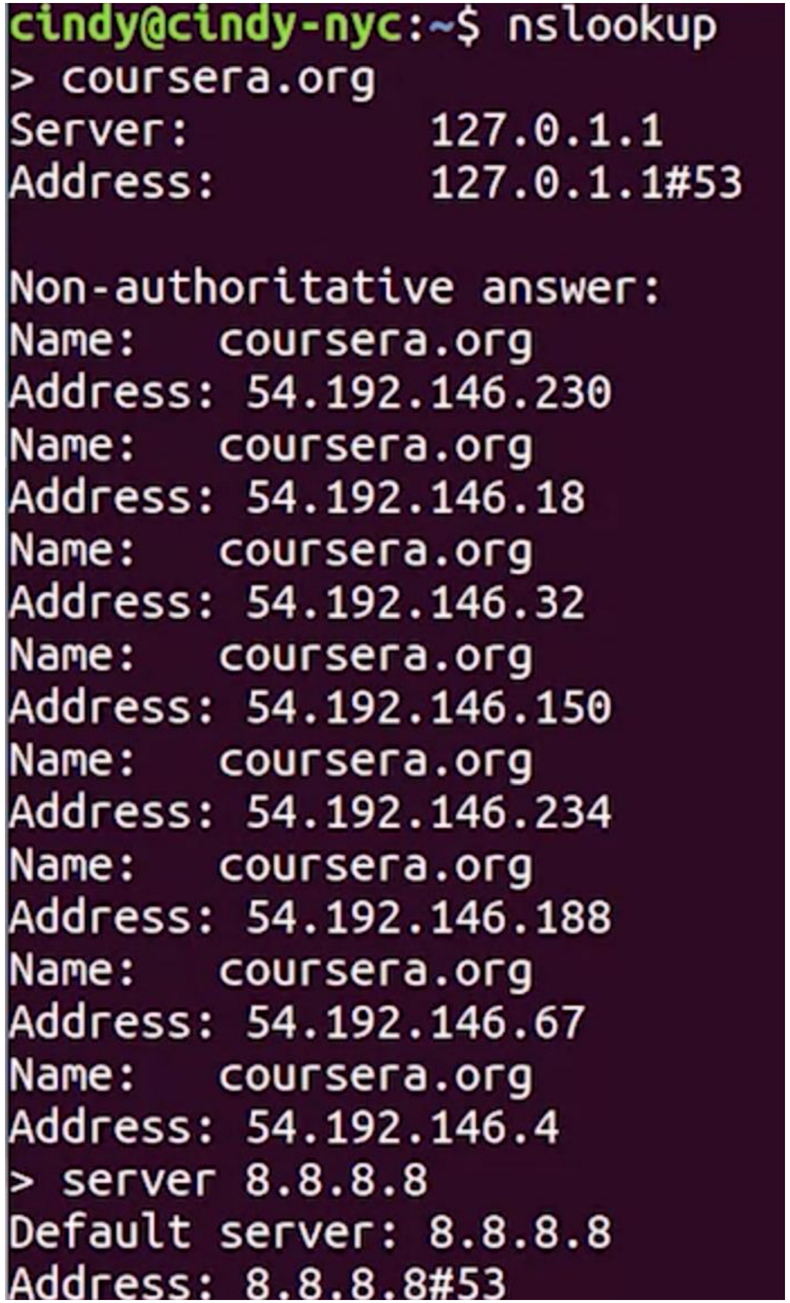
# Name Resolution Tools

Nslookup – works for all operation system (check the lookup table)



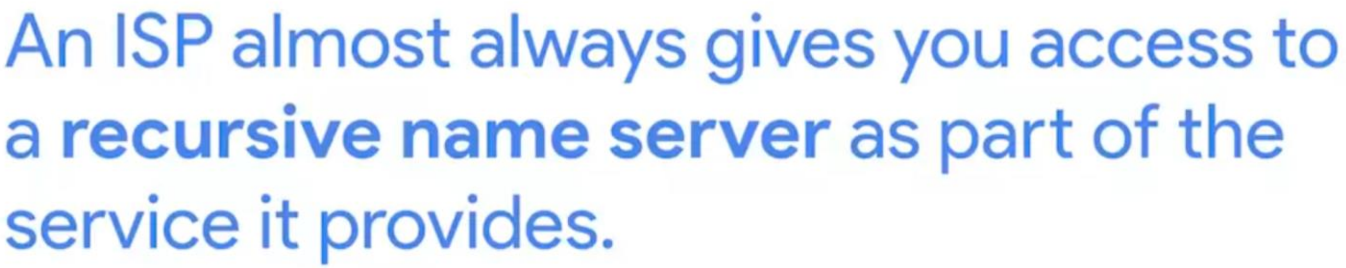
```
cindy@cindy-nyc:~$ nslookup twitter.com
Server:         127.0.1.1
Address:        127.0.1.1#53

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

For activating the interactive model, just type nslookup and hit the enter

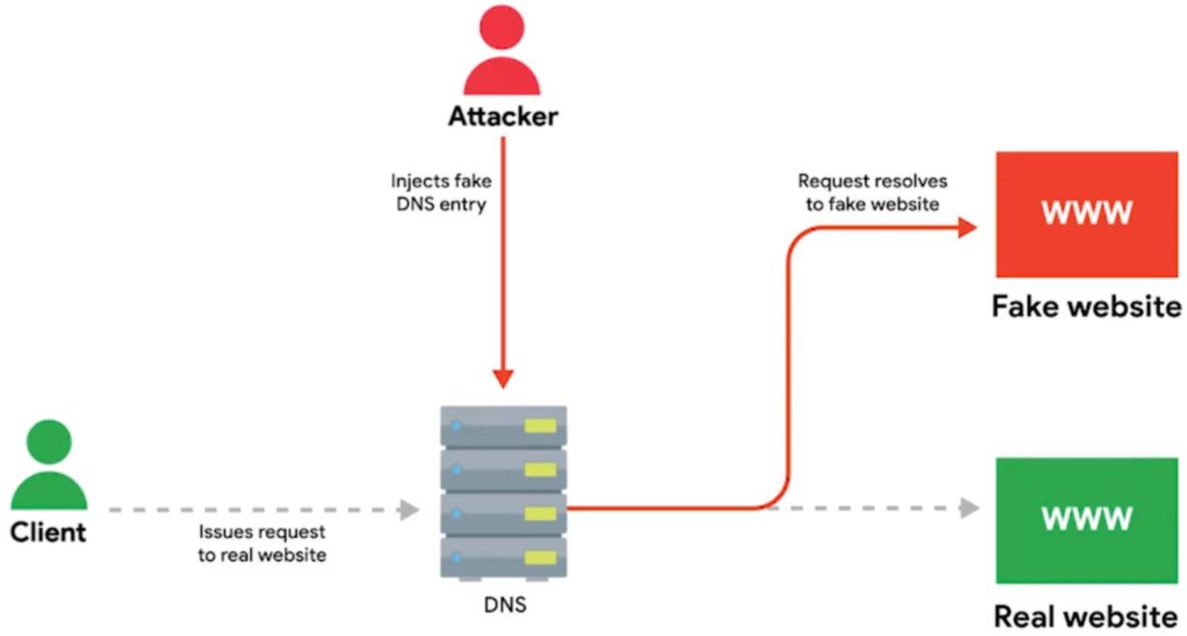


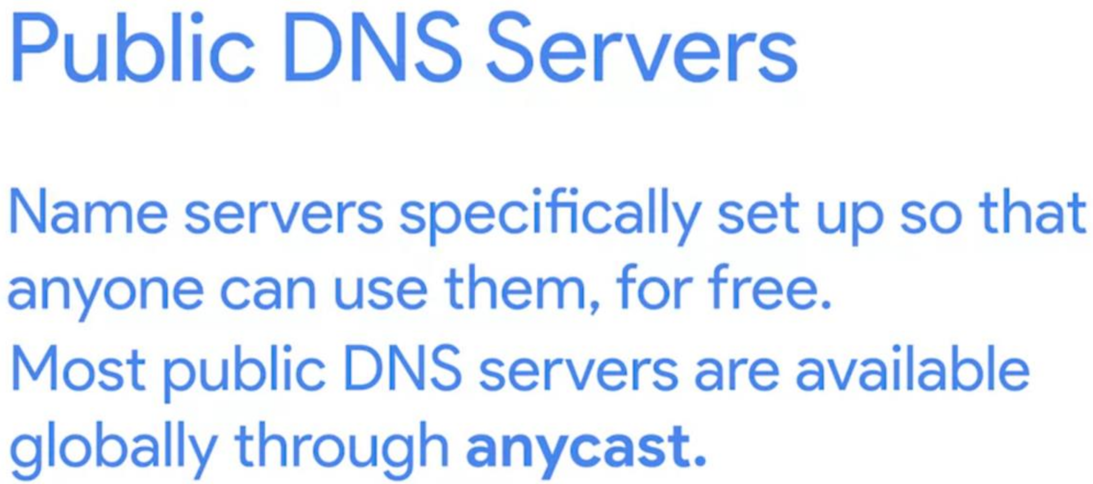
```
cindy@cindy-nyc:~$ nslookup
> coursera.org
Server:         127.0.1.1
Address:        127.0.1.1#53

Non-authoritative answer:
Name:   coursera.org
Address: 54.192.146.230
Name:   coursera.org
Address: 54.192.146.18
Name:   coursera.org
Address: 54.192.146.32
Name:   coursera.org
Address: 54.192.146.150
Name:   coursera.org
Address: 54.192.146.234
Name:   coursera.org
Address: 54.192.146.188
Name:   coursera.org
Address: 54.192.146.67
Name:   coursera.org
Address: 54.192.146.4
> server 8.8.8.8
Default server: 8.8.8.8
Address: 8.8.8.8#53
```

# Public DNS Servers



An ISP almost always gives you access to a **recursive name server** as part of the service it provides.



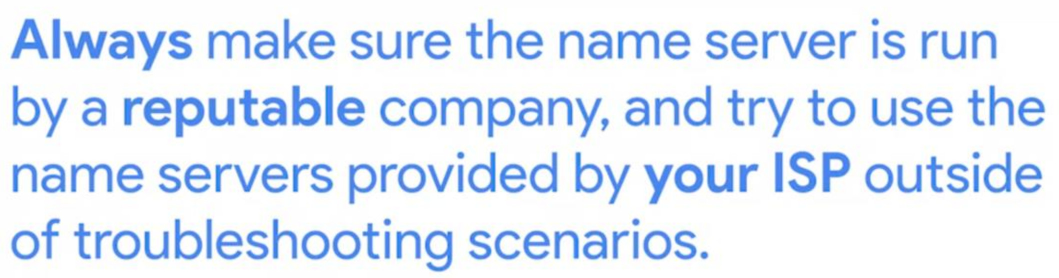


## Public DNS Servers

Name servers specifically set up so that anyone can use them, for free.

Most public DNS servers are available globally through **anycast.**

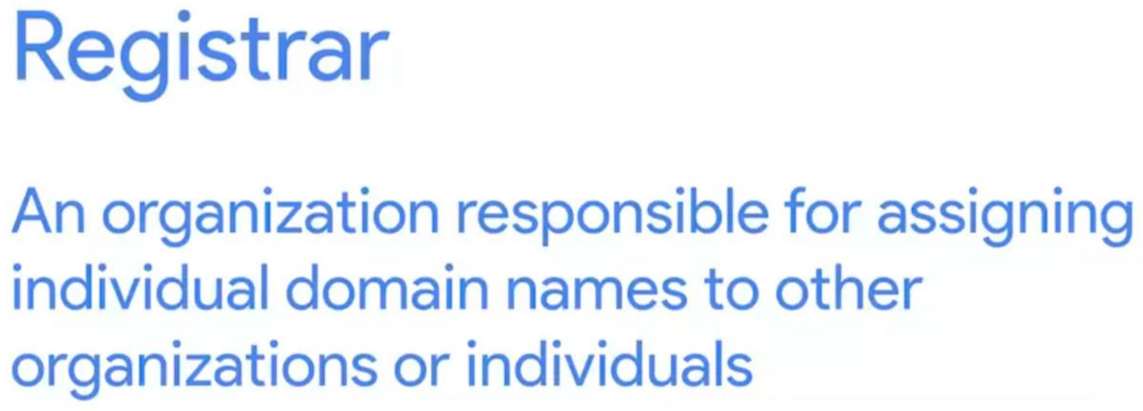
Google host their free DNS server at 8.8.8.8 and 8.8.8.4



**Always** make sure the name server is run by a **reputable** company, and try to use the name servers provided by **your ISP** outside of troubleshooting scenarios.

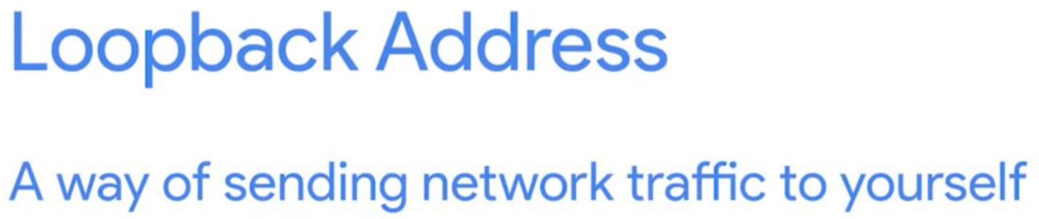
Internet Service Provider (ISP)

# DNS Registration and Expiration



## Registrar

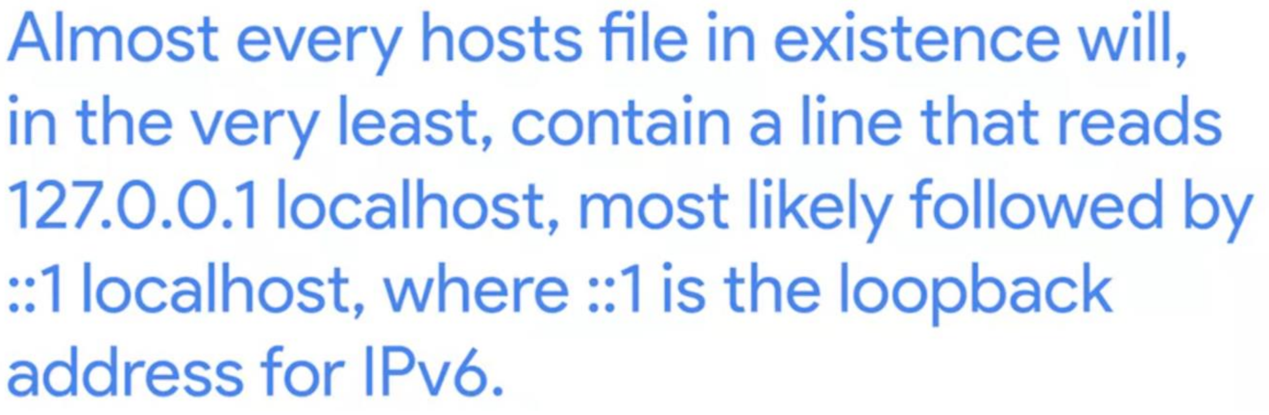
An organization responsible for assigning individual domain names to other organizations or individuals



## Loopback Address

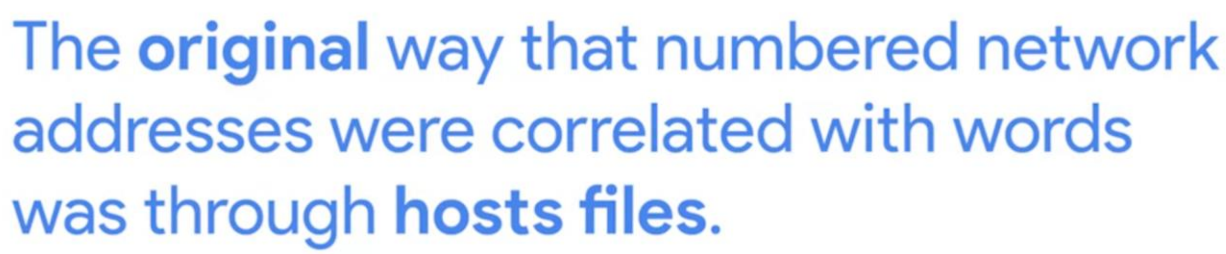
A way of sending network traffic to yourself

For IPV4, loopback address is 127.0.0.1.

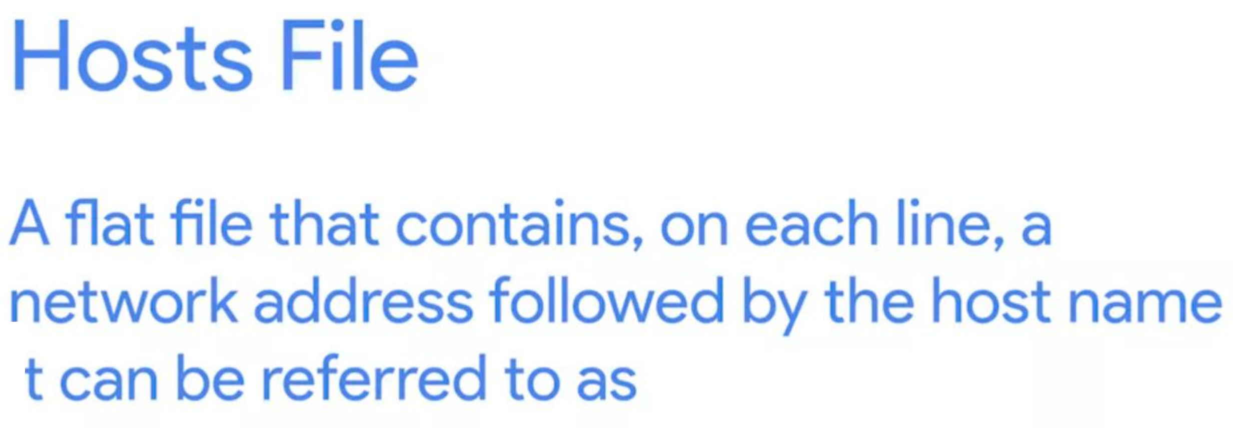


Almost every hosts file in existence will, in the very least, contain a line that reads 127.0.0.1 localhost, most likely followed by ::1 localhost, where ::1 is the loopback address for IPv6.

## Hosts Files



The **original** way that numbered network addresses were correlated with words was through **hosts files.**



## Hosts File

A flat file that contains, on each line, a network address followed by the host name t can be referred to as

Like system variable, for example: 1.2.3.4 Webserver, then you could type ping Webserver, it would automatically connect to 1.2.3.4

Hosts files are a popular way for computer viruses to disrupt and redirect users' traffic.

# What is The Cloud?

Realize it by hardware virtualization

## Cloud Computing

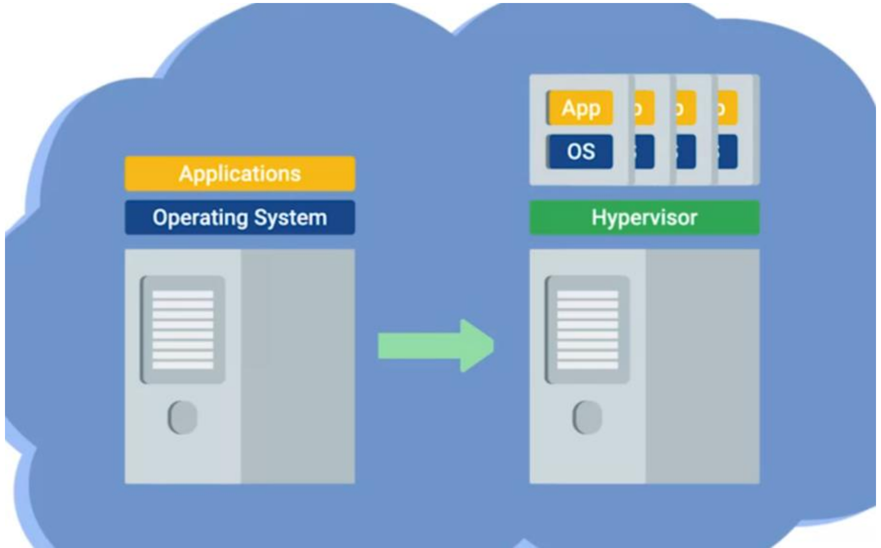
A technological approach where computing resources are provisioned in a shareable way, so that lots of users get what they need, when they need it

## Virtualization

A single physical machine, called a host, could run many individual virtual instances, called guests

## Hypervisor

A piece of software that runs and manages virtual machines, while also offering these guests a virtual operating platform that's indistinguishable from actual hardware



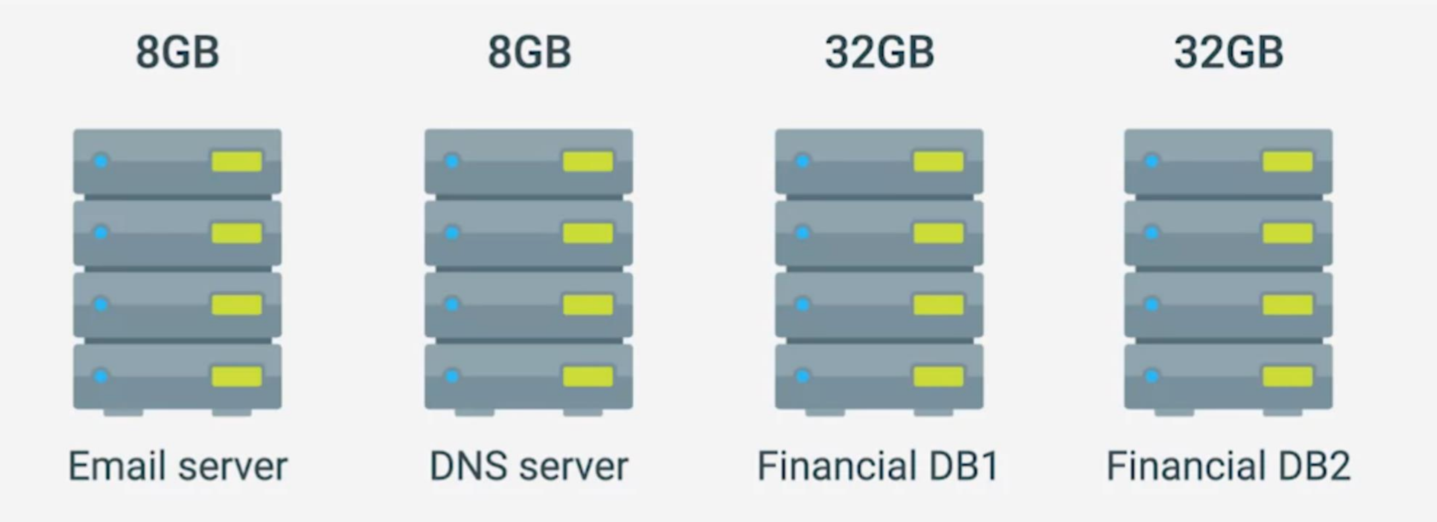
Without cloud (hardware virtualization) if we want to host a server, we need

1. Email server (works on windows)
2. DNS Server (Works on Linux)
3. Database Server (Works on Linux)
4. Backup Server for Database

Total 80 GB Ram are required, but half of them are rarely used

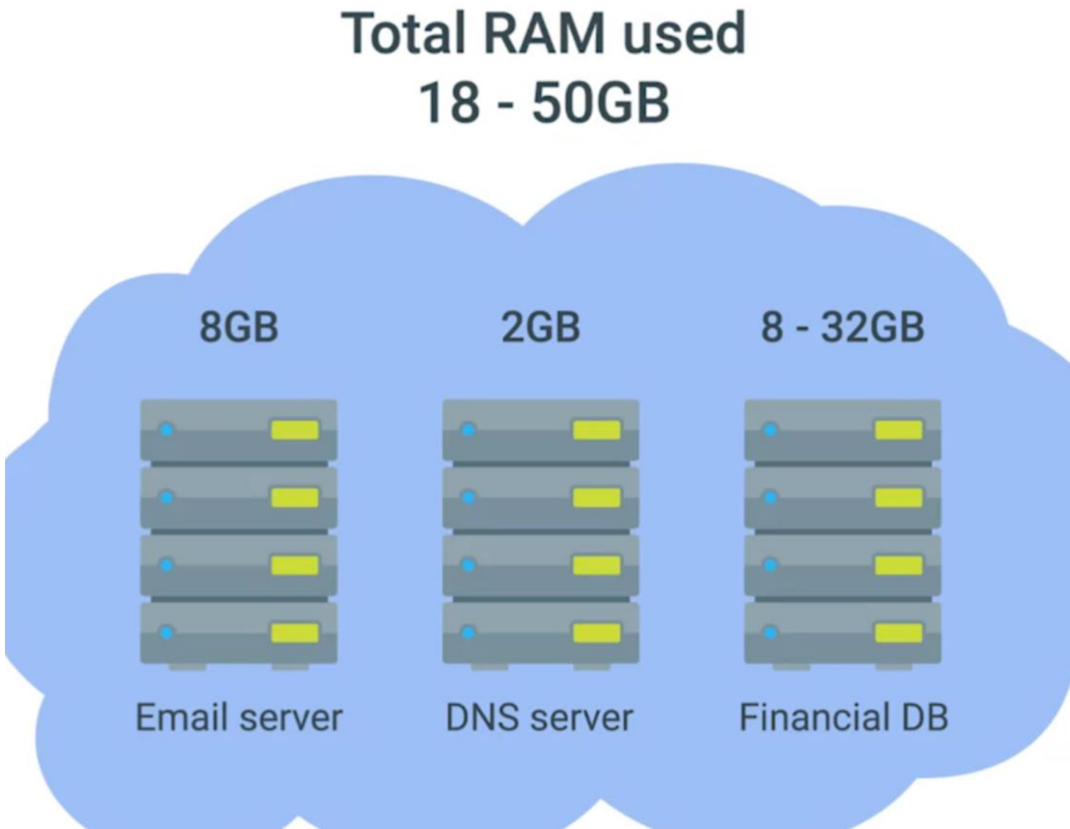
## Private cloud

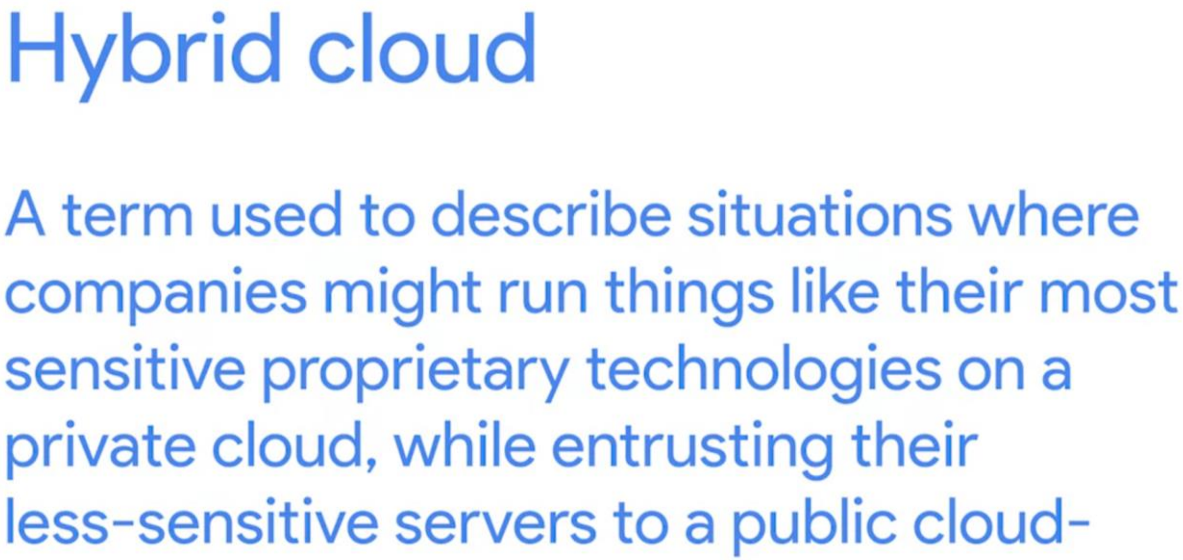
Used by a single large corporation and generally physically hosted on its own premises



## Public cloud

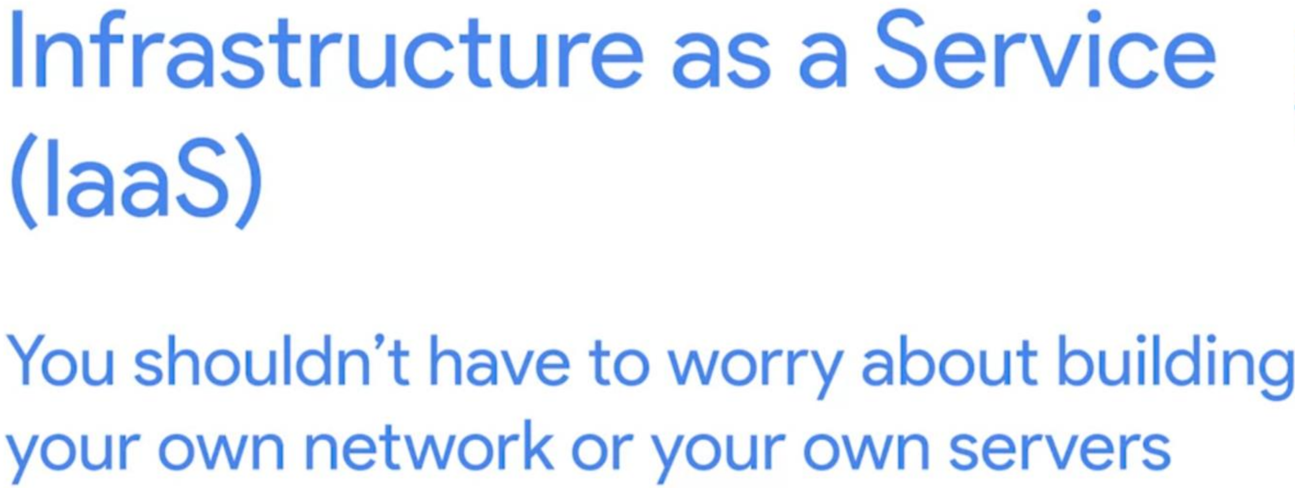
A large cluster of machines run by another company





## Hybrid cloud

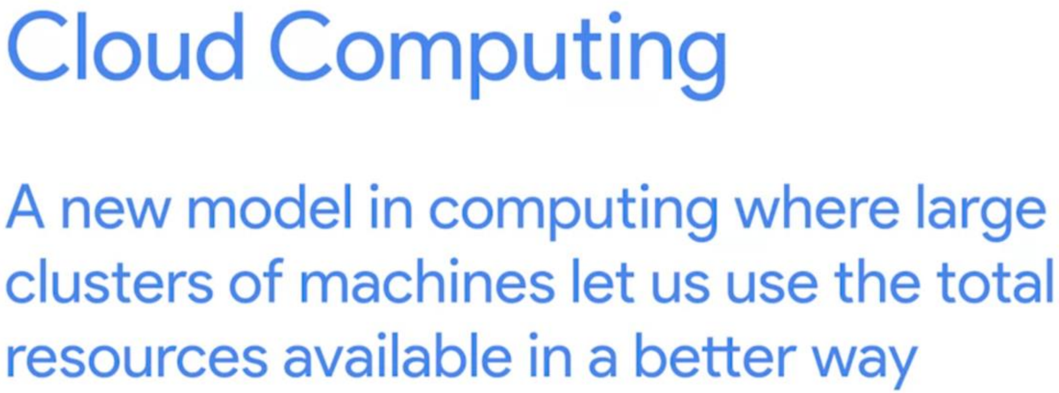
A term used to describe situations where companies might run things like their most sensitive proprietary technologies on a private cloud, while entrusting their less-sensitive servers to a public cloud-



## Infrastructure as a Service (IaaS)

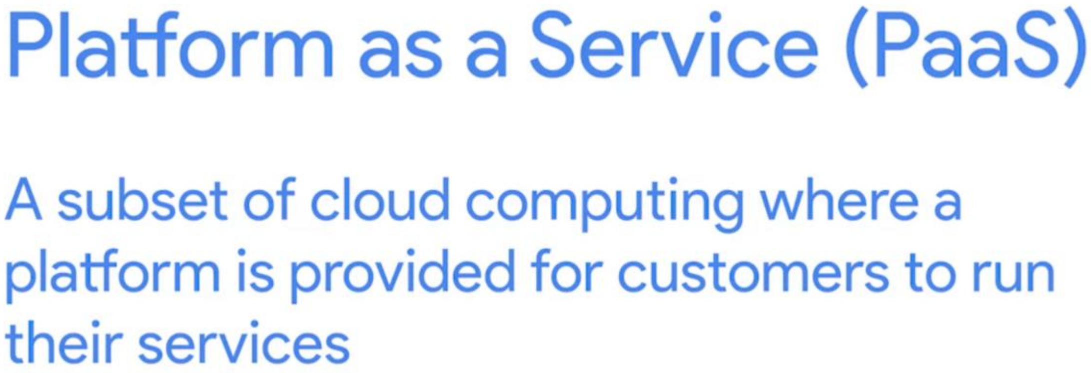
You shouldn’t have to worry about building your own network or your own servers

1. Platform as a Service (Paas)
2. Software as a Service (Saas)



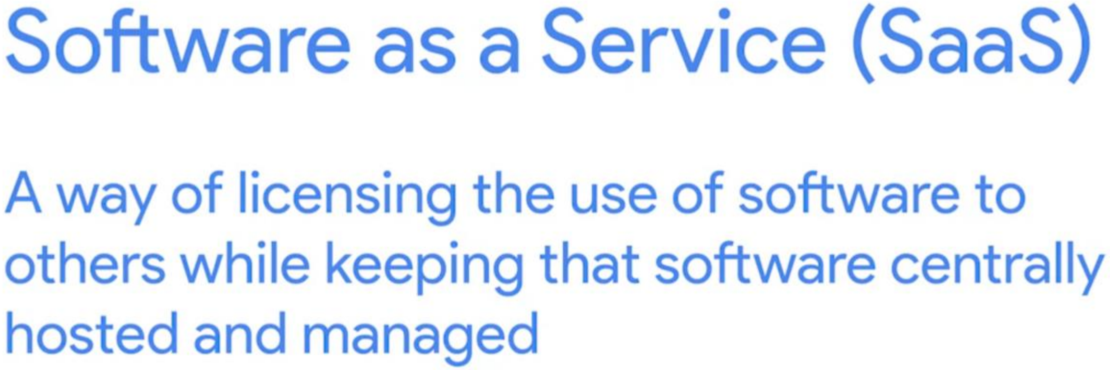
## Cloud Computing

A new model in computing where large clusters of machines let us use the total resources available in a better way



## Platform as a Service (PaaS)

A subset of cloud computing where a platform is provided for customers to run their services



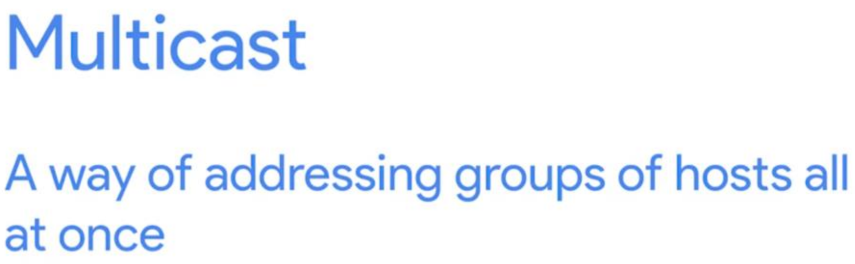
## Software as a Service (SaaS)

A way of licensing the use of software to others while keeping that software centrally hosted and managed

# Cloud Storage

Use the cloud providers’ protocol to keep user’s data

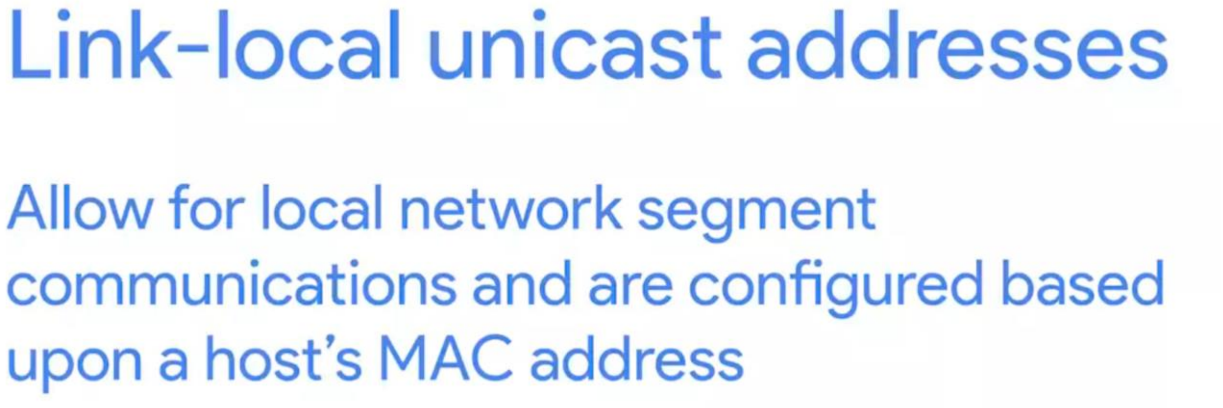
1. Secure
2. Accessible
3. Available



## Multicast

A way of addressing groups of hosts all at once

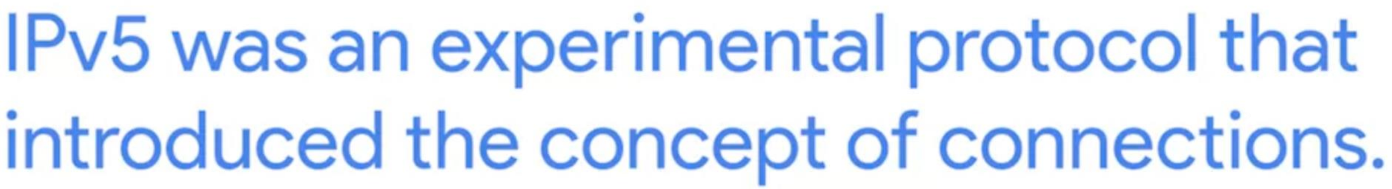
FE80:: reserves for link-local unicast



## Link-local unicast addresses

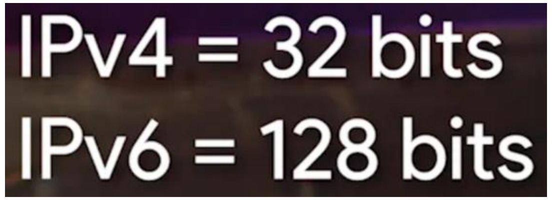
Allow for local network segment communications and are configured based upon a host’s MAC address

# IPV6



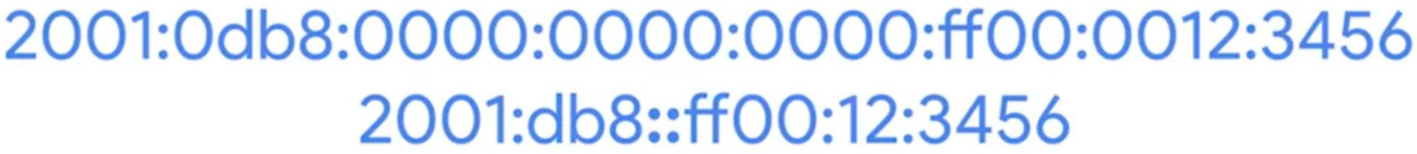
IPv5 was an experimental protocol that introduced the concept of connections.

1. Remove the leading 0
2. Replace “all 0” as “::”



IPv4 = 32 bits

IPv6 = 128 bits

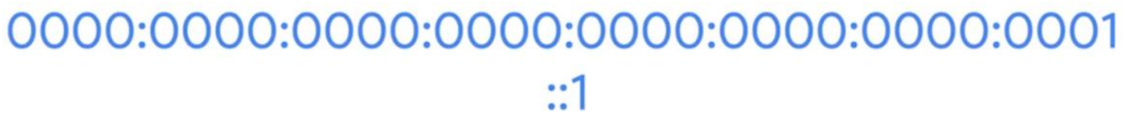


2001:0db8:0000:0000:0000:ff00:0012:3456

2001:db8::ff00:12:3456

Every address start with FF00: is reserved for multicast

Lookback address:



0000:0000:0000:0000:0000:0000:0000:0001

::1

# IPv6 Headers

2001:0db8:0000:0000:0000:ff00:0012:3456

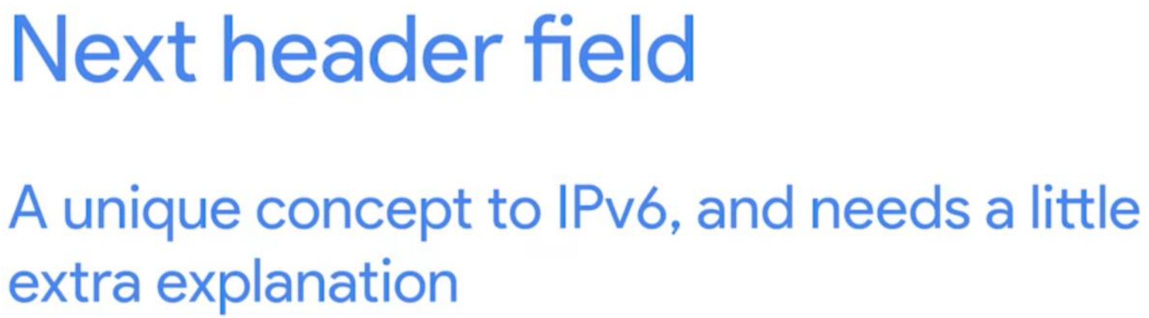
Network ID Host ID

## Hop limit field

An 8-bit field that’s identical in purpose to the TTL field in an IPv4 header

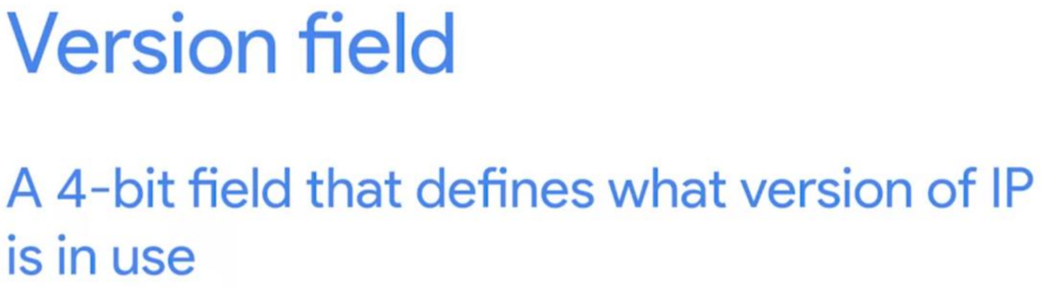
## Payload length field

A 16-bit field that defines how long the data payload section of the datagram is



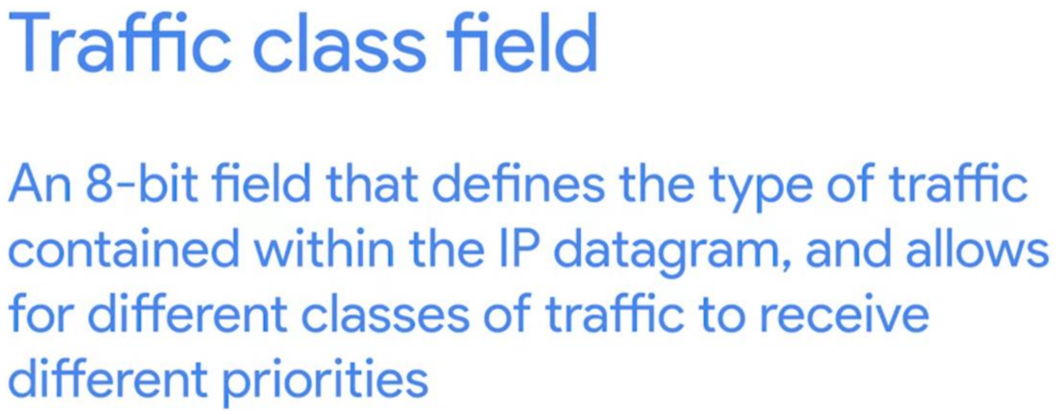
## Next header field

A unique concept to IPv6, and needs a little extra explanation



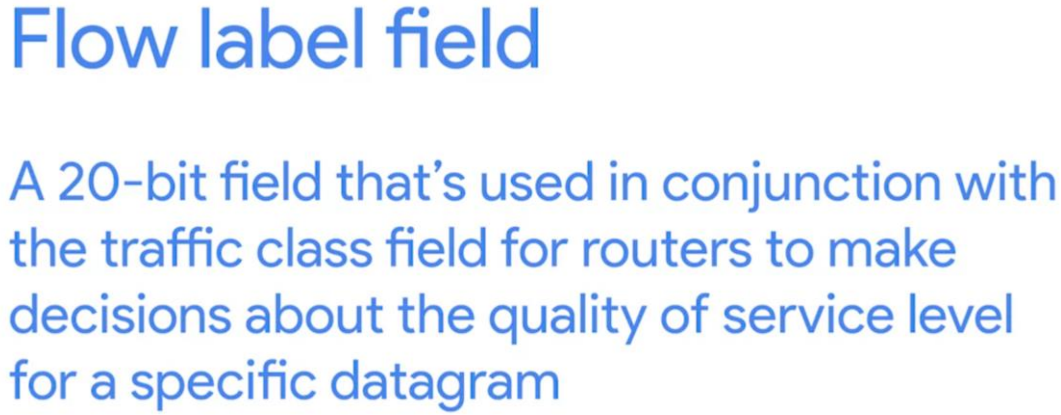
## Version field

A 4-bit field that defines what version of IP is in use



## Traffic class field

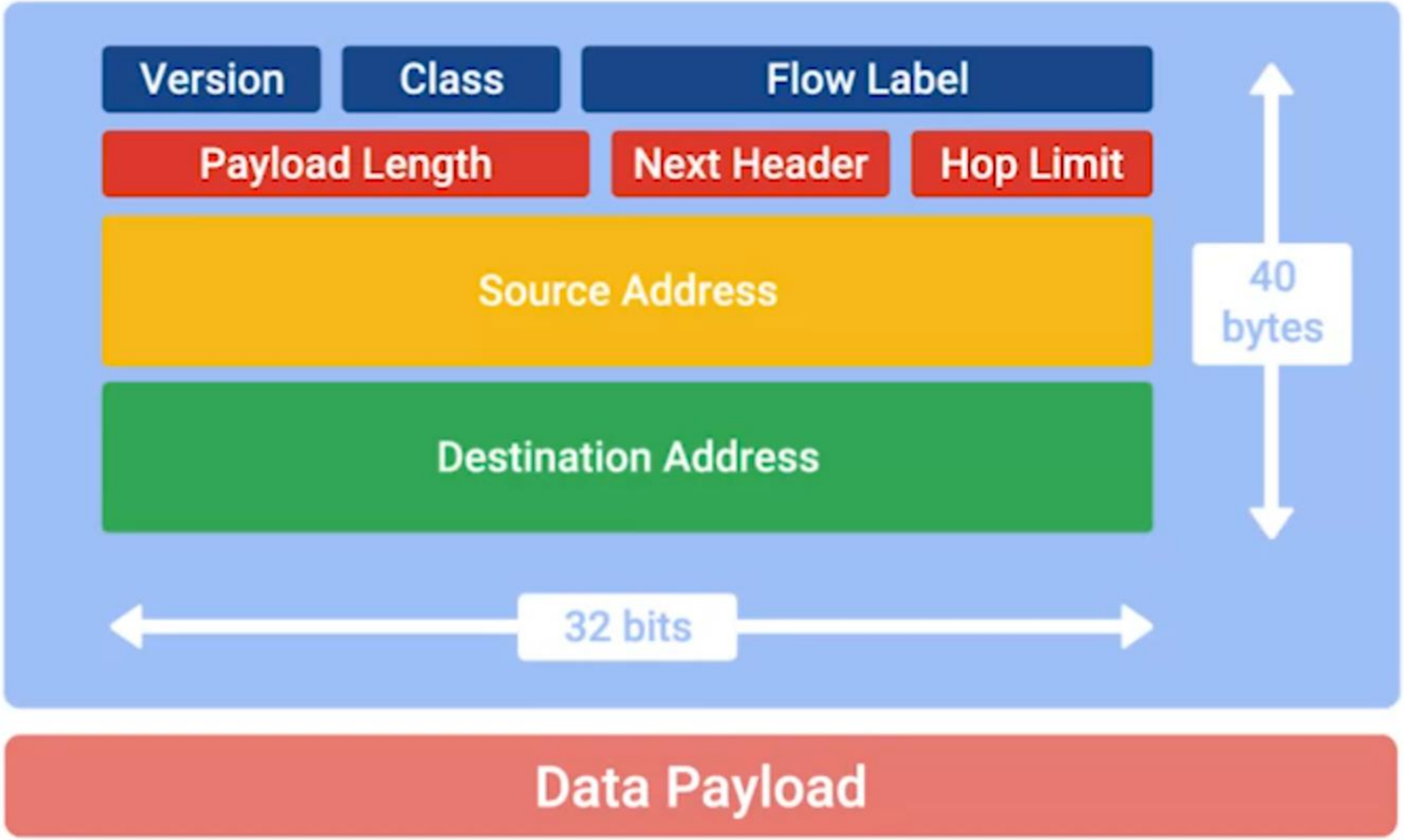
An 8-bit field that defines the type of traffic contained within the IP datagram, and allows for different classes of traffic to receive different priorities



## Flow label field

A 20-bit field that’s used in conjunction with the traffic class field for routers to make decisions about the quality of service level for a specific datagram

# IPv6 Headers



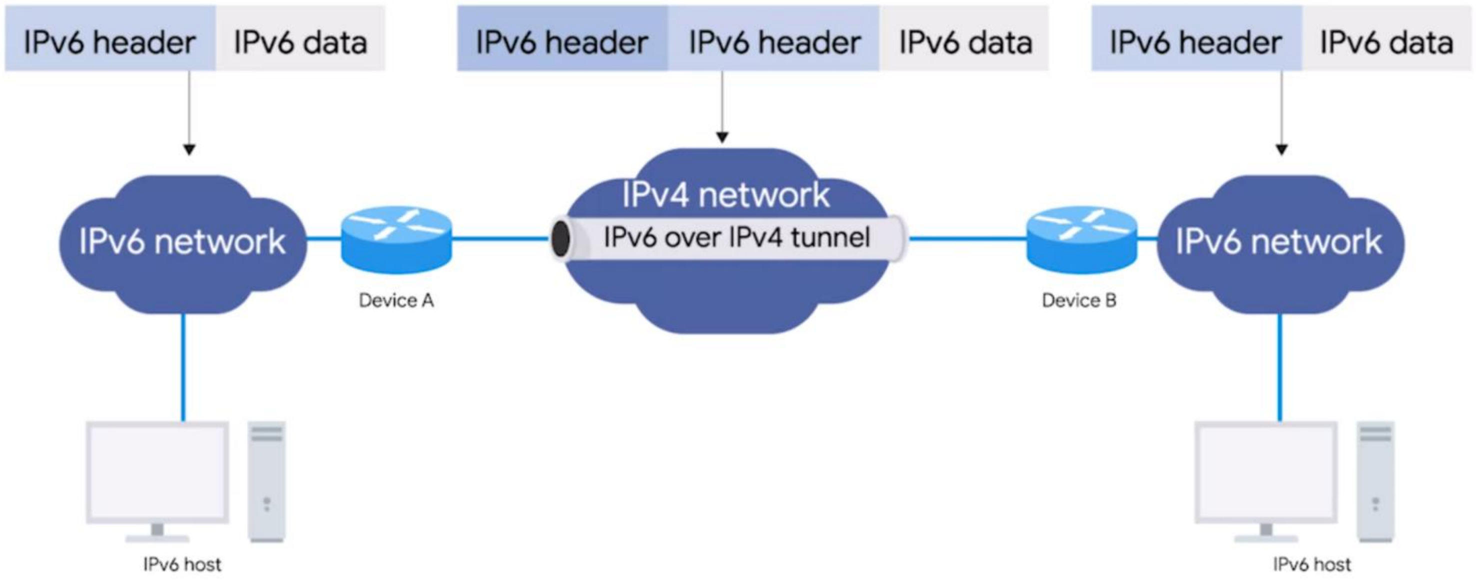
# IPV4 and IPV6 Co-exist

IPV4 Mapped Address

192.168.1.1 = 0:0:0:0:0:ffff:d1ad:35a7

## IPv6 tunnels

Servers take incoming IPv6 traffic and encapsulate it within traditional IPv4 datagram



## IPv6 tunnel broker

Companies that provide IPv6 tunneling endpoints for you, so you don't have to introduce additional equipment to your network

# Supplemental Reading for IPv6 and IPv4 Harmony

- While IPv6 adoption becomes more widespread, it'll need a way to travel over the old IPv4 remnants of the Internet backbone. The primary way this is achieved today is through **IPv6 tunnels**. IPv6 tunnels are conceptually pretty simple. They consist of IPv6 tunnel servers on either end of a connection. These IPv6 tunnel servers take incoming IPv6 traffic and encapsulate it within traditional IPv4 datagrams. This is then delivered across the IPv4 Internet space where it's received by another IPv6 tunnel server. That server performs the de-encapsulation and passes the IPv6 traffic further along in the network.
- There are a lot of competing protocols to be used for these kinds of IPv6 tunnels. Since this is still a new and evolving space, it's not clear who the winner will be. Some of the main competitors are **6in4**, **Tunnel Setup Protocol**, and **Anything in Anything (AYIYA)**.