**Exercise 1: try 3 websites and report the ip addresses**

Asos.com – error message

Youtube.com - 216.58.212.238

Substack.com - error message

**Exercise 2 – 3 domain names to access traceroute info**

Google

enter the website or ip address: google.co.uk

1 \* \* \*

2 \* \* \*

3 \* \* \*

4 \* \* \*

5 \* \* \*

6 \* \* \*

7 \* \* \*

8 \* \* \*

9 ber01s14-in-f3.1e100.net (216.58.213.3) 8.369 ms 4.513 ms 4.243 ms

Youtube

1 \* \* \*

2 \* \* \*

3 \* \* \*

4 \* \* \*

5 \* \* \*

6 \* \* \*

7 \* \* \*

8 \* \* \*

9 ams16s21-in-f14.1e100.net (216.58.212.206) 26.935 ms 12.044 ms 10.957 ms

Asos

1 \* \* \*

2 \*traceroute: wrote asos.com 40 chars, ret=-1

\*traceroute: wrote asos.com 40 chars, ret=-1

\*

3 traceroute: wrote asos.com 40 chars, ret=-1

\*traceroute: wrote asos.com 40 chars, ret=-1

\*traceroute: wrote asos.com 40 chars, ret=-1

\*

4 traceroute: wrote asos.com 40 chars, ret=-1

\*traceroute: wrote asos.com 40 chars, ret=-1

\* \*

5 \* \* \*

6 \* \* \*

7 \* \* \*

8 \* \* \*

9 \* \* \*

10 a2-21-15-57.deploy.static.akamaitechnologies.com (2.21.15.57) 14.469 ms 13.291 ms 15.186 ms

Exercise 3 – analyse traceroutes

Youtube

* First 8 lines refer to hops, the asterisks means the exact info was blocked to likely prevent probing through timeouts (lack of response) or nodes being unreachable
* They may not be confiugured for ICMP messages
* Hop 9 is successful, with ams16s21-in-f14.1e100.net being the destination server, the ip address being 216.58.212.206, likely in Amsterdam. The packets reached the destination
* 26.935 is time for first packet, 12.044 for the second and 10.957 for the third
* These are roundtrips, quick roundtrips mean the networks functioning quickly an there is no congestion

Google

* First 8 lines refer to hops, the asterisks means the exact info was blocked to likely prevent probing through timeouts (lack of response) or nodes being unreachable
* They may not be confiugured for ICMP messages
* Hop 9 is successful, ber01s14-in-f3.1e100.net being the destination server, the ip address being 2216.58.213.3, likely in Berlin. The packets reached the destination
* Packet 1 took 8.369 ms, packet 2 took 4.513 ms, packet 3 took 4.243 ms
* These are roundtrips, quick roundtrips mean the networks functioning quickly an there is no congestion

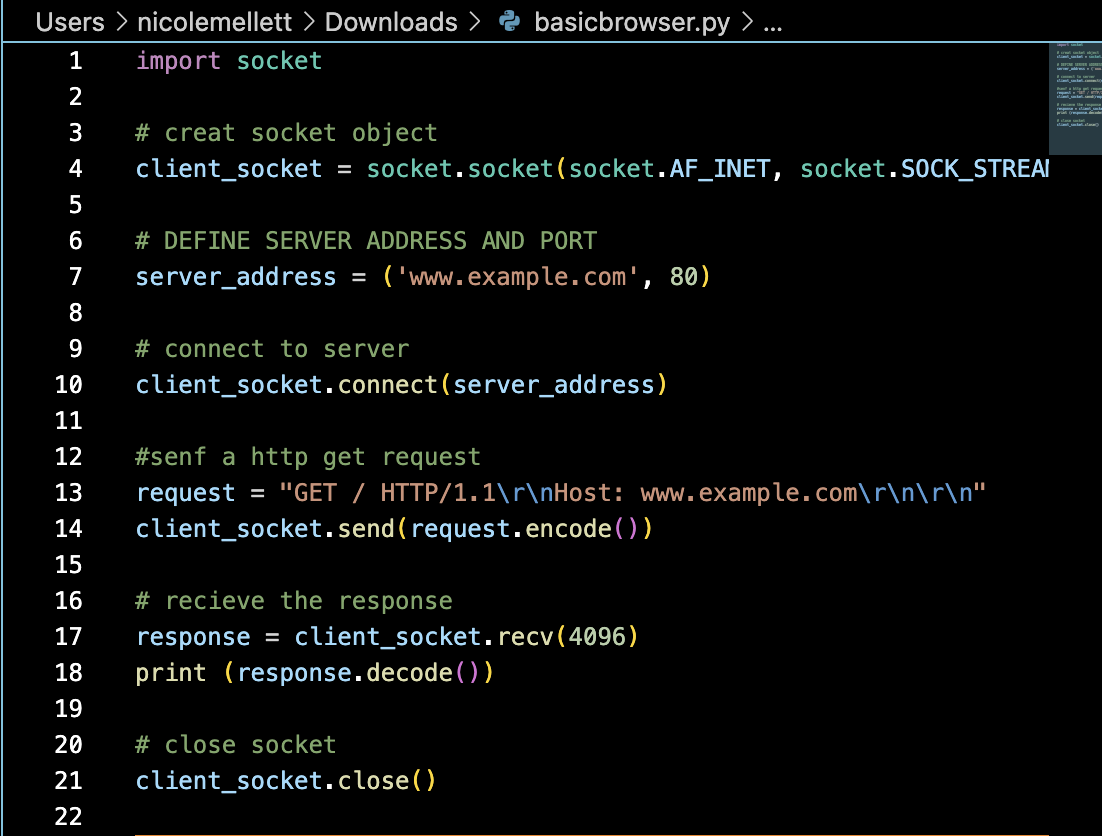
Asos

* 10 hops between routers
* Despite different outputs to previous traceroutes, the first 9 hops are still unreachable due to privacy, configuration or nodes being unreachable
* This could be a blockage of any icmp ot udp protocol
* Traffic may not be expected to reply to thse hop requests
* The ip address of final router is 2.21.15.57 as shown on hop 10
* The destination server is a2-21-15-57.deploy.static.akamaitechnologies.com
* This is the company Akamai Technologies, a popular cdn provider
* This is content delivery as images, styling css and javascript
* The packets delivered in 14.469 ms, 13.291 ms, 15.186 ms roundtime – to Akamais server and back
* Low latency suggests global optimizized servers

Exercise 4:

* All timeout hops imply there may be network problems and congestion
* We cannot see the times between hops, further implying there may be issues here
* The routers could be slow amd overwhelmed
* The first hops may be provided by my end’s isp
* If these are consistently timing out, this may suggest bottlenecks in my isps routing

Simulating a basic browser in Python



* Uses socket library in python

Creation of socket

* Socket is network communication endpoint
* Script creates a socket object
* It specifies that IP and TCP will be used
* TCP delivers data reliably and in the correct order

Connection to server

* Script connects socket to server
* This is at [www.example.com](http://www.example.com) at port 80
* Port 80 is standard port for HTTP
* Establishes connection between python script, which is client side, and the web server

Send HTTP Request

* Scipt creates http get request
* Tells web server wants the main web page
* Host header specifies domain name
* \r\n represents carriage return and line feed characters, which separate lines in http headers
* Request is sent over socket to server

Receive response

* Script waits for the server to respond
* Recv() method receives data sent by server
* This is the http response
* Stores it in response variable

Print reponse

* Script prints received response
* Response is html content of [www.example.com](http://www.example.com) web page
* Also prints http headers that provide info on the response

Close socket

* Script closes socket
* Terminates connection to server

Python requests library

* Socket module can be consuming for everyday tasks
* Requests library simplies http requests and is widely used in industry
* Easier syntax for http requests
* Auto handling of headers, cookies and character encoding

Http Requests type

Common requests

GET – fetches read-only data from server

POST – sends data to a server to create or change a resource

PUT – updates existing resource on server, requires full updated data to be sent in request body

DELETE – deletes resource from server, doesn’t require a request body

POST

* Send data to crate / update a resource
* Submits data as part of request unlike a GET request
* Used in web applications like user registration and login forms, uploading files, submitting forms and creating records in a database

POST over GET

* Data sent in request body not the url
* Keeps url clean and allows for larger data sets
* More secure for sensitive data as info is not stored in browser history or visible in url logs