1. **Traceroute** [**www.microsoft.com**](http://www.microsoft.com)
2. Results:

Traceroute has started…

traceroute to e13678.dspb.akamaiedge.net (23.59.156.241), 64 hops max, 72 byte packets

1 10.0.0.1 (10.0.0.1) 11.232 ms 7.720 ms 6.191 ms

2 70.71.64.1 (70.71.64.1) 20.251 ms 21.352 ms 16.538 ms

3 rc1st-tge0-14-0-2-1.vc.shawcable.net (64.59.150.245) 22.232 ms 20.341 ms 18.897 ms

4 rc1bb-tge0-0-0-28.vc.shawcable.net (66.163.69.197) 22.430 ms 21.705 ms 19.833 ms

5 rc3ar-tge0-6-0-16.ed.shawcable.net (66.163.76.42) 29.278 ms 25.956 ms 25.911 ms

6 a23-59-156-241.deploy.static.akamaitechnologies.com (23.59.156.241) 28.880 ms 15.707 ms 14.971 ms

B) The ISP networks that the Traceroute packets pass through are Shaw Communications Inc and Akamai Technologies.

C) Based on the traceroute results, the data packet’s IP address destination (23.59.156.241) is located in Cambridge, Boston Massachusetts. This is because the ISP that hosts Microsoft servers is Akamai Technologies which is not located in Seattle.

**Traceroute www.apple.com**

1. Results:

Traceroute has started…

traceroute to e6858.dsce9.akamaiedge.net (69.192.85.46), 64 hops max, 72 byte packets

1 10.0.0.1 (10.0.0.1) 6.323 ms 9.929 ms 5.051 ms

2 70.71.64.1 (70.71.64.1) 16.310 ms 18.933 ms 15.289 ms

3 rc1st-tge0-14-0-2-1.vc.shawcable.net (64.59.150.245) 17.048 ms 16.142 ms 19.936 ms

4 rc1bb-tge0-0-0-28.vc.shawcable.net (66.163.69.197) 15.410 ms 17.459 ms 16.755 ms

5 a69-192-85-46.deploy.static.akamaitechnologies.com (69.192.85.46) 20.482 ms 20.743 ms 17.292 ms

1. The ISP networks that the Traceroute packets pass through are Shaw Communications Inc and Akamai Technologies.
2. Based on the traceroute results, the data packet’s destination IP address (69.192.85.46) is located in Cambridge Boston Massachusetts. The ISP that hosts Apple servers is the same as the ISP that hosts Microsoft servers which is Akamai Technologies.

**Traceroute** [**www.google.com**](http://www.google.com)

1. Results:

Traceroute has started…

traceroute to www.google.com (216.58.193.68), 64 hops max, 72 byte packets

1 10.0.0.1 (10.0.0.1) 7.029 ms 8.507 ms 6.014 ms

2 70.71.64.1 (70.71.64.1) 17.725 ms 16.171 ms 15.246 ms

3 rc1st-tge0-14-0-2-1.vc.shawcable.net (64.59.150.245) 16.862 ms 17.032 ms 19.565 ms

4 rc2wt-be50-1.wa.shawcable.net (66.163.70.106) 23.521 ms 20.800 ms 22.962 ms

5 72.14.242.90 (72.14.242.90) 19.785 ms 22.737 ms 24.564 ms

6 \* \* \*

7 sea15s07-in-f68.1e100.net (216.58.193.68) 29.780 ms 24.569 ms 17.919 ms

1. The ISP networks that the Traceroute packets pass through are Shaw Communications Inc and Google Inc
2. Based on the traceroute results, the data packet’s destination IP address (216.58.193.68) is located in Draper, Utah.

Unlike Apple and Microsoft, Google hosts its own servers. Google servers are located in Draper, Utah. Apple and Microsoft both use Akamai Technologies’ services to host their network servers. Apple and Microsoft servers are both located in Cambridge Boston Massachusetts where Akamai Technologies is located.

1. A) RFC 2616 for HTTP 1.1

Section numbers for:

Persistent HTTP: section 8.1

Caching: section 13

Commands (GET and etc.): section 9

Transport Layer Protocol: section 1.1

B) A persistent connection with pipelining is when a client sends multiple requests to a server without waiting for a response. The server has to respond to those requests in the same order in which the requests were sent to the server. If the first pipeline attempt fails, the client must retry their connection. On this retry, the client must ensure that their connection is persistent first before pipelining. If the server closes the connection between the server and the client, the client must resend their requests if the server closes the connection before it gives a response to those requests.

1. Python source code:

import random  
  
class node:  
 def \_\_init\_\_(self, peer\_id,key, succ\_table):  
 self.peer\_id = peer\_id  
 self.key = key  
 self.succ\_table = succ\_table  
  
  
def chord\_dht(inputfile, outputfile):  
  
 S = None  
 N = None  
 M = None  
 hashed\_ids = list()  
 hashed\_ids\_string = list()  
 hashed\_keys = list()  
 hashed\_keys\_string = list()  
 end = list()  
 end\_string = list()  
 node\_list = list()  
  
 with open(inputfile, "r") as input\_file:  
  
 S = int(input\_file.readline())  
 N = int(input\_file.readline())  
 M = int(input\_file.readline())  
  
 hashed\_ids\_string = input\_file.readline().split(',')  
 for id in hashed\_ids\_string:  
 hashed\_ids.insert(int(id),int(id))  
  
 hashed\_keys\_string = input\_file.readline().split(',')  
 for key in hashed\_keys\_string:  
 hashed\_keys.insert(int(key),int(key))  
  
 end\_string = input\_file.readline().split(',')  
 for end\_value in end\_string:  
 end.append(int(end\_value))  
  
 temp\_hashed\_ids = hashed\_ids  
  
 for i in range(0,pow(2,M)):  
 while len(temp\_hashed\_ids) != 0:  
 node\_id = random.sample(temp\_hashed\_ids,1)[0]  
 aNode = node(node\_id, None,None)  
 temp\_hashed\_ids.remove(node\_id)  
 node\_list.append(aNode)  
  
  
 for key in hashed\_keys:  
 min = 999999  
 for node\_item in node\_list:  
 difference = key - node\_item.peer\_id  
 if abs(difference) < min:  
 min = abs(difference)  
  
 for node\_item in node\_list:  
 if node\_item.peer\_id == key - min:  
 node\_item.key = key  
  
 for node\_item in node\_list:  
 table = list()  
 for i in range(0,M+1):  
 table.append([])  
 node\_item.succ\_table = table  
  
 ring = [None] \* int(S+1)  
  
  
 for node\_item in node\_list:  
 ring[node\_item.peer\_id] = node\_item  
  
  
 for node\_item in ring:  
 if node\_item != None:  
 for i in range(0,M+1):  
 node\_item.succ\_table[i].append(i)  
 node\_item.succ\_table[i].append((node\_item.peer\_id + pow(2,i)) % int(S+1))  
  
 k = 1  
 while(1):  
 if ring[((node\_item.peer\_id + pow(2,i))) % int(S+1)] != None:  
 if ring[((node\_item.peer\_id + pow(2,i))) % int(S+1)].peer\_id == (node\_item.peer\_id + pow(2, i)) % int(S + 1):  
 node\_item.succ\_table[i].append(ring[((node\_item.peer\_id + pow(2,i))) % int(S+1)].peer\_id)  
 break  
  
 elif ring[((node\_item.peer\_id + pow(2,i))+k) % int(S+1)] != None:  
 node\_item.succ\_table[i].append(ring[((node\_item.peer\_id + pow(2,i))+k) % int(S+1)].peer\_id)  
 break  
 k = k+1  
  
  
 with open(outputfile, "w") as output\_file:  
 for node\_item in ring:  
 if node\_item != None and node\_item.key != None:  
 output\_file.write(str(node\_item.peer\_id) + "\n")  
 for row in node\_item.succ\_table:  
 for item in row:  
 output\_file.write(str(item) + " ")  
 output\_file.write("\n")  
  
chord\_dht("test.txt","output\_file.txt")

1. Wireshark Lab 1: Packet Sniffer
2. 3 protocols that appear are DNS, SSDP and ICMPv6
3. It took 38.039455 – 37.940903 = 0.098552 seconds for the OK reply to be received after the HTTP GET message was sent
4. The Internet address of gaia.cs.umass.edu is 128.119.245.12. The Internet address of my computer is 10.0.0.32

4. GET MESSAGE) Frame 315: 466 bytes on wire (3728 bits), 466 bytes captured (3728 bits) on interface 0

Interface id: 0 (en0)

Interface name: en0

Encapsulation type: Ethernet (1)

Arrival Time: Feb 26, 2019 01:43:31.268722000 PST

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1551174211.268722000 seconds

[Time delta from previous captured frame: 0.000399000 seconds]

[Time delta from previous displayed frame: 0.000000000 seconds]

[Time since reference or first frame: 37.940903000 seconds]

Frame Number: 315

Frame Length: 466 bytes (3728 bits)

Capture Length: 466 bytes (3728 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:tcp:http]

[Coloring Rule Name: HTTP]

[Coloring Rule String: http || tcp.port == 80 || http2]

Ethernet II, Src: Apple\_2b:05:e7 (f0:18:98:2b:05:e7), Dst: ArrisGro\_b1:67:d2 (10:56:11:b1:67:d2)

Destination: ArrisGro\_b1:67:d2 (10:56:11:b1:67:d2)

Address: ArrisGro\_b1:67:d2 (10:56:11:b1:67:d2)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Source: Apple\_2b:05:e7 (f0:18:98:2b:05:e7)

Address: Apple\_2b:05:e7 (f0:18:98:2b:05:e7)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 10.0.0.32, Dst: 128.119.245.12

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Total Length: 452

Identification: 0x0000 (0)

Flags: 0x4000, Don't fragment

0... .... .... .... = Reserved bit: Not set

.1.. .... .... .... = Don't fragment: Set

..0. .... .... .... = More fragments: Not set

...0 0000 0000 0000 = Fragment offset: 0

Time to live: 64

Protocol: TCP (6)

Header checksum: 0xb990 [validation disabled]

[Header checksum status: Unverified]

Source: 10.0.0.32

Destination: 128.119.245.12

Transmission Control Protocol, Src Port: 60956, Dst Port: 80, Seq: 1, Ack: 1, Len: 400

Source Port: 60956

Destination Port: 80

[Stream index: 7]

[TCP Segment Len: 400]

Sequence number: 1 (relative sequence number)

[Next sequence number: 401 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

1000 .... = Header Length: 32 bytes (8)

Flags: 0x018 (PSH, ACK)

000. .... .... = Reserved: Not set

...0 .... .... = Nonce: Not set

.... 0... .... = Congestion Window Reduced (CWR): Not set

.... .0.. .... = ECN-Echo: Not set

.... ..0. .... = Urgent: Not set

.... ...1 .... = Acknowledgment: Set

.... .... 1... = Push: Set

.... .... .0.. = Reset: Not set

.... .... ..0. = Syn: Not set

.... .... ...0 = Fin: Not set

[TCP Flags: ·······AP···]

Window size value: 2058

[Calculated window size: 131712]

[Window size scaling factor: 64]

Checksum: 0x8968 [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

TCP Option - No-Operation (NOP)

Kind: No-Operation (1)

TCP Option - No-Operation (NOP)

Kind: No-Operation (1)

TCP Option - Timestamps: TSval 1413767222, TSecr 1646332276

Kind: Time Stamp Option (8)

Length: 10

Timestamp value: 1413767222

Timestamp echo reply: 1646332276

[SEQ/ACK analysis]

[iRTT: 0.097675000 seconds]

[Bytes in flight: 400]

[Bytes sent since last PSH flag: 400]

[Timestamps]

[Time since first frame in this TCP stream: 0.098074000 seconds]

[Time since previous frame in this TCP stream: 0.000399000 seconds]

TCP payload (400 bytes)

Hypertext Transfer Protocol

GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n

[Expert Info (Chat/Sequence): GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n]

[GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1\r\n]

[Severity level: Chat]

[Group: Sequence]

Request Method: GET

Request URI: /wireshark-labs/INTRO-wireshark-file1.html

Request Version: HTTP/1.1

Host: gaia.cs.umass.edu\r\n

Upgrade-Insecure-Requests: 1\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8\r\n

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_3) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0.3 Safari/605.1.15\r\n

Accept-Language: en-ca\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

\r\n

[Full request URI: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html]

[HTTP request 1/1]

[Response in frame: 317]

OK MESSAGE) Frame 317: 504 bytes on wire (4032 bits), 504 bytes captured (4032 bits) on interface 0

Interface id: 0 (en0)

Interface name: en0

Encapsulation type: Ethernet (1)

Arrival Time: Feb 26, 2019 01:43:31.367274000 PST

[Time shift for this packet: 0.000000000 seconds]

Epoch Time: 1551174211.367274000 seconds

[Time delta from previous captured frame: 0.000004000 seconds]

[Time delta from previous displayed frame: 0.098552000 seconds]

[Time since reference or first frame: 38.039455000 seconds]

Frame Number: 317

Frame Length: 504 bytes (4032 bits)

Capture Length: 504 bytes (4032 bits)

[Frame is marked: False]

[Frame is ignored: False]

[Protocols in frame: eth:ethertype:ip:tcp:http:data-text-lines]

[Coloring Rule Name: HTTP]

[Coloring Rule String: http || tcp.port == 80 || http2]

Ethernet II, Src: ArrisGro\_b1:67:d2 (10:56:11:b1:67:d2), Dst: Apple\_2b:05:e7 (f0:18:98:2b:05:e7)

Destination: Apple\_2b:05:e7 (f0:18:98:2b:05:e7)

Address: Apple\_2b:05:e7 (f0:18:98:2b:05:e7)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Source: ArrisGro\_b1:67:d2 (10:56:11:b1:67:d2)

Address: ArrisGro\_b1:67:d2 (10:56:11:b1:67:d2)

.... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

.... ...0 .... .... .... .... = IG bit: Individual address (unicast)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 128.119.245.12, Dst: 10.0.0.32

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Total Length: 490

Identification: 0xa7b9 (42937)

Flags: 0x4000, Don't fragment

0... .... .... .... = Reserved bit: Not set

.1.. .... .... .... = Don't fragment: Set

..0. .... .... .... = More fragments: Not set

...0 0000 0000 0000 = Fragment offset: 0

Time to live: 46

Protocol: TCP (6)

Header checksum: 0x23b1 [validation disabled]

[Header checksum status: Unverified]

Source: 128.119.245.12

Destination: 10.0.0.32

Transmission Control Protocol, Src Port: 80, Dst Port: 60956, Seq: 1, Ack: 401, Len: 438

Source Port: 80

Destination Port: 60956

[Stream index: 7]

[TCP Segment Len: 438]

Sequence number: 1 (relative sequence number)

[Next sequence number: 439 (relative sequence number)]

Acknowledgment number: 401 (relative ack number)

1000 .... = Header Length: 32 bytes (8)

Flags: 0x018 (PSH, ACK)

000. .... .... = Reserved: Not set

...0 .... .... = Nonce: Not set

.... 0... .... = Congestion Window Reduced (CWR): Not set

.... .0.. .... = ECN-Echo: Not set

.... ..0. .... = Urgent: Not set

.... ...1 .... = Acknowledgment: Set

.... .... 1... = Push: Set

.... .... .0.. = Reset: Not set

.... .... ..0. = Syn: Not set

.... .... ...0 = Fin: Not set

[TCP Flags: ·······AP···]

Window size value: 235

[Calculated window size: 30080]

[Window size scaling factor: 128]

Checksum: 0xebca [unverified]

[Checksum Status: Unverified]

Urgent pointer: 0

Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

TCP Option - No-Operation (NOP)

Kind: No-Operation (1)

TCP Option - No-Operation (NOP)

Kind: No-Operation (1)

TCP Option - Timestamps: TSval 1646332377, TSecr 1413767222

Kind: Time Stamp Option (8)

Length: 10

Timestamp value: 1646332377

Timestamp echo reply: 1413767222

[SEQ/ACK analysis]

[iRTT: 0.097675000 seconds]

[Bytes in flight: 438]

[Bytes sent since last PSH flag: 438]

[Timestamps]

[Time since first frame in this TCP stream: 0.196626000 seconds]

[Time since previous frame in this TCP stream: 0.000004000 seconds]

TCP payload (438 bytes)

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]

[HTTP/1.1 200 OK\r\n]

[Severity level: Chat]

[Group: Sequence]

Response Version: HTTP/1.1

Status Code: 200

[Status Code Description: OK]

Response Phrase: OK

Date: Tue, 26 Feb 2019 09:43:31 GMT\r\n

Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/5.4.16 mod\_perl/2.0.10 Perl/v5.16.3\r\n

Last-Modified: Tue, 26 Feb 2019 06:59:01 GMT\r\n

ETag: "51-582c697975da4"\r\n

Accept-Ranges: bytes\r\n

Content-Length: 81\r\n

[Content length: 81]

Keep-Alive: timeout=5, max=100\r\n

Connection: Keep-Alive\r\n

Content-Type: text/html; charset=UTF-8\r\n

\r\n

[HTTP response 1/1]

[Time since request: 0.098552000 seconds]

[Request in frame: 315]

File Data: 81 bytes

Line-based text data: text/html (3 lines)

<html>\n

Congratulations! You've downloaded the first Wireshark lab file!\n

</html>\n

Wireshark Lab 2 (HTTP):

1)Both my browser and the server are running HTTP 1.1

2) Accept-Language: en-us

3) My IP address is 10.0.0.32 and the server’s is 128.119.245.12

4) The status code returned was HTTP/1.1 200 OK (text/html)

5) The HTML file I am retrieving was last modified 25 Feb 2019 18:13:01 GMT

6) There are 126 bytes of content

7) No, the raw data will contain all the headers

8) No

9) Yes the server explicitly returned the contents of the file because its contents are shown in the Line-based data text field

10)Yes, the information that follows is Mon, 25 Feb 2019 18:32:02 GMT which is when the file was last modified from the previous GET request

11) The status code returned is “HTTP/1.1 304 not modified. The browser loaded the contents from its cache therefore the server did not return anything

12) There was 1 HTTP GET request messages sent by my browser

13)There are 4 segments

14) 200 OK

15) No

16) There were 3 HTTP GET requests. Their addresses were: 128.119.245.12, 165.193.123.218, 134.241.6.82

17) Yes we can tell. If we check the TCP ports, we can see if our files were downloaded serially or in parallel

18) Status code: 401, Phrase: Authorization required

19) The new field is Authorization: Basic ISRoOXN0SWRlbnRIOm5ldHOSvcPSz\r\n