HTTP Questions (1-4)

1. Choose 5 HTTP status codes and describe each one.

The 5 HTTP status codes are:

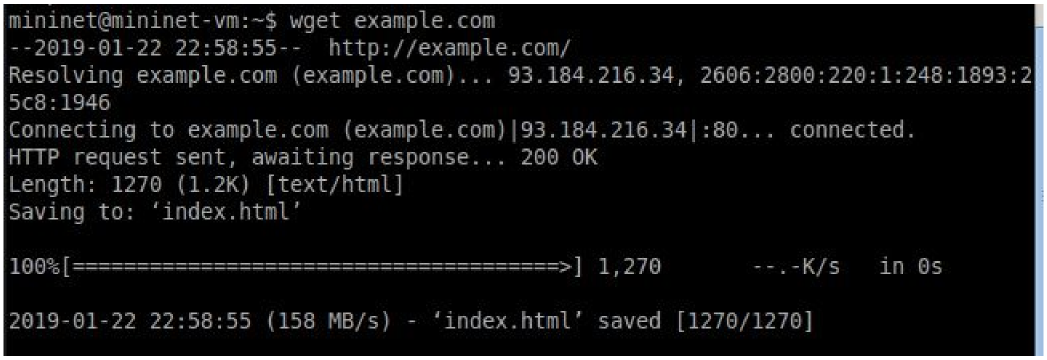
- 101 which is switching protocols
- 2XX which means the client's request was successfully read, understood, and accepted
- 3XX which means redirection, indicating that further action is required by user agent to fulfill the request
- 4XX is client error, in which the client itself runs into an error
- 5XX is server error, which means the server knows there is an error, or it is incapable of fulfilling the request

2. List the 8 HTTP 1.1 methods and explain what they do.

The 8 HTTP 1.1 methods are

- Safe and Idempotent methods
- Options
- Get
- Head
- Post
- Put
- Delete
- Trace
- Connect

3. Use *wget* on *example.com* to view the last modified date of the webpage. What was the HTTP return status given and what command was used to do this? (The command should not download the file! Hint: Look into the wget man page.)



```
mininet@mininet-vm:~$ wget example.com
--2019-01-22 22:58:55--  http://example.com/
Resolving example.com (example.com)... 93.184.216.34, 2606:2800:220:1:248:1893:2
5c8:1946
Connecting to example.com (example.com)|93.184.216.34|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1270 (1.2K) [text/html]
Saving to: ‘index.html’

100%[======================================>] 1,270       --.-K/s   in 0s

2019-01-22 22:58:55 (158 MB/s) - ‘index.html’ saved [1270/1270]
```

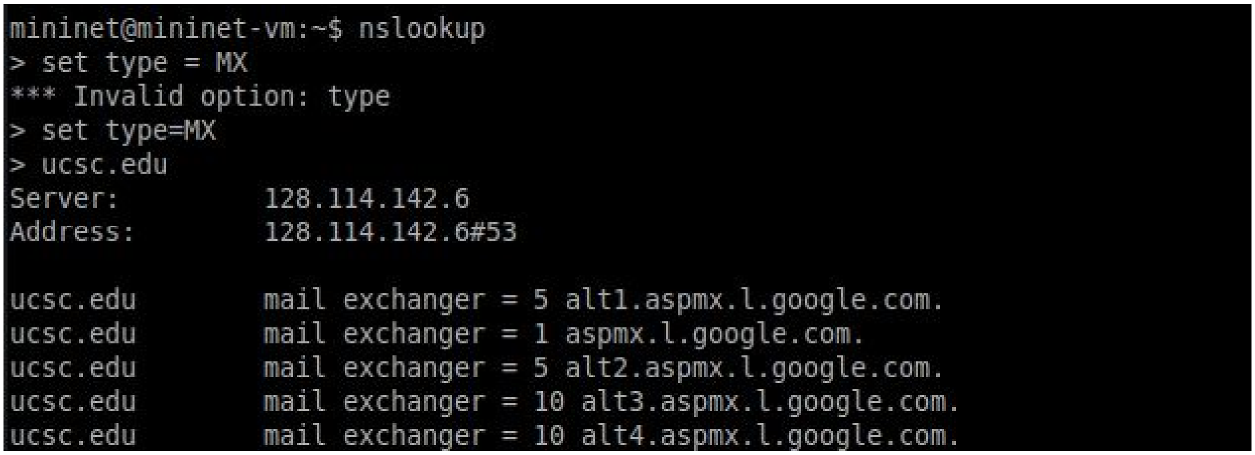
The HTTP status return code, was 200, which meant it was successful.

4. Look up the *telnet* command. Use *telnet* to connect to *towel.blinkenlights.nl*. What does this telnet server do?

The server plays Star Wars in ASCII texts.

DNS Questions (5-6)

5. In your own words describe what a DNS resource record (RR) is. Now using the command line tool *nslookup* find the *MX* resource record of *ucsc.edu*. What does this resource record mean?



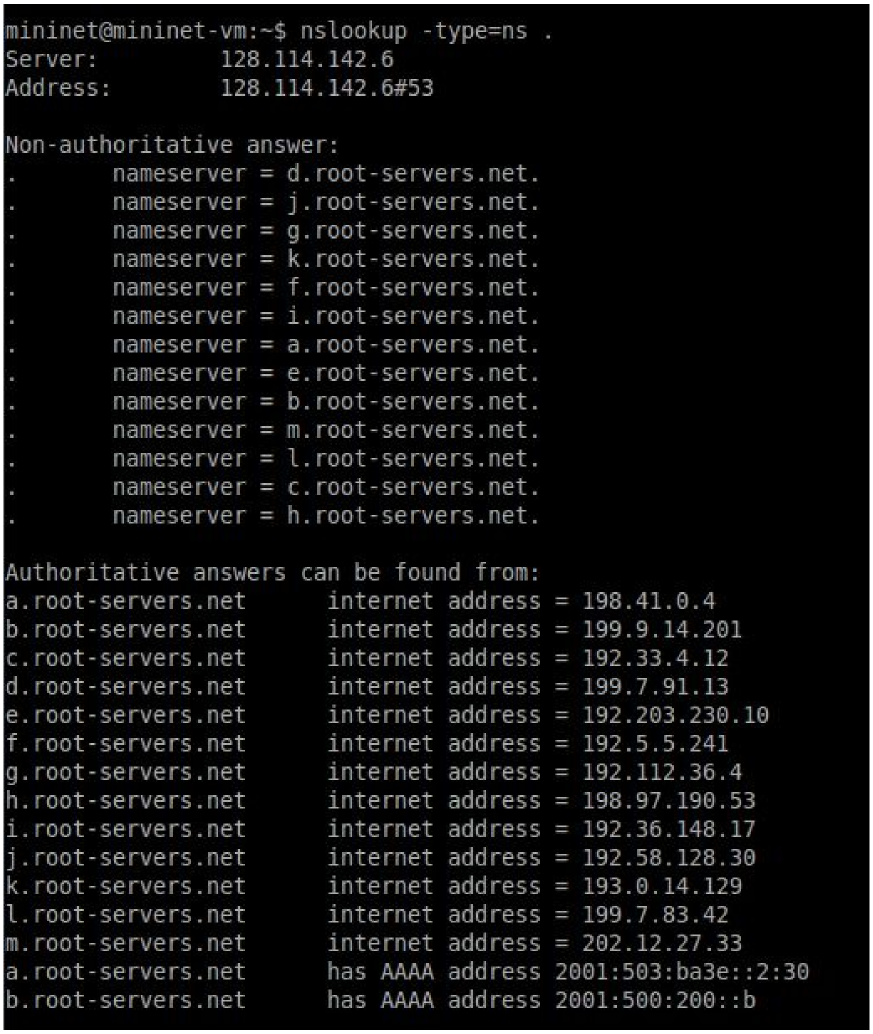
```
mininet@mininet-vm:~$ nslookup
> set type = MX
*** Invalid option: type
> set type=MX
> ucsc.edu
Server:         128.114.142.6
Address:        128.114.142.6#53

ucsc.edu        mail exchanger = 5 alt1.aspmx.l.google.com.
ucsc.edu        mail exchanger = 1 aspmx.l.google.com.
ucsc.edu        mail exchanger = 5 alt2.aspmx.l.google.com.
ucsc.edu        mail exchanger = 10 alt3.aspmx.l.google.com.
ucsc.edu        mail exchanger = 10 alt4.aspmx.l.google.com.
```

It is a data type in the Domain Name System which is sent across networks in a text format during zone transfers, they are basic building blocks of host names and IP information.

6. What does the command *nslookup type=ns .* do*?* Explain its output. (Note: the . is part of the command!) **[There is a picture on the next page, that shows the output of the command]**

It simulates a list of all the name servers that are authoritative to that domain. The bottom output is a list of



```
mininet@mininet-vm:~$ nslookup -type=ns .
Server:         128.114.142.6
Address:        128.114.142.6#53

Non-authoritative answer:
.       nameserver = d.root-servers.net.
.       nameserver = j.root-servers.net.
.       nameserver = g.root-servers.net.
.       nameserver = k.root-servers.net.
.       nameserver = f.root-servers.net.
.       nameserver = i.root-servers.net.
.       nameserver = a.root-servers.net.
.       nameserver = e.root-servers.net.
.       nameserver = b.root-servers.net.
.       nameserver = m.root-servers.net.
.       nameserver = l.root-servers.net.
.       nameserver = c.root-servers.net.
.       nameserver = h.root-servers.net.

Authoritative answers can be found from:
a.root-servers.net      internet address = 198.41.0.4
b.root-servers.net      internet address = 199.9.14.201
c.root-servers.net      internet address = 192.33.4.12
d.root-servers.net      internet address = 199.7.91.13
e.root-servers.net      internet address = 192.203.230.10
f.root-servers.net      internet address = 192.5.5.241
g.root-servers.net      internet address = 192.112.36.4
h.root-servers.net      internet address = 198.97.190.53
i.root-servers.net      internet address = 192.36.148.17
j.root-servers.net      internet address = 192.58.128.30
k.root-servers.net      internet address = 193.0.14.129
l.root-servers.net      internet address = 199.7.83.42
m.root-servers.net      internet address = 202.12.27.33
a.root-servers.net      has AAAA address 2001:503:ba3e::2:30
b.root-servers.net      has AAAA address 2001:500:200::b
```

TCP Questions (1-4)

1. How can multiple application services running on a single machine with a single IP address be uniquely identified?

A connection in TCP has 4 unique identifications, source and destination port, source and destination IP address. For example, if connecting to the same web server twice from our client, will have 2 distinct source ports from our perspective and destination ports for the server.

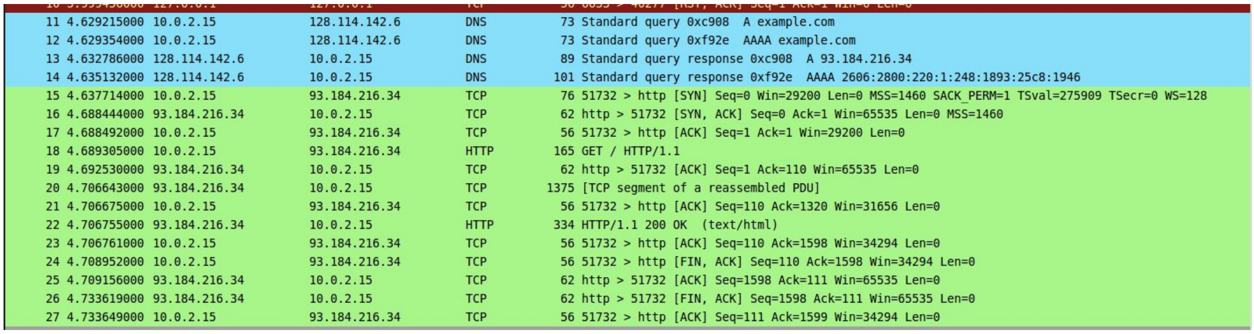
2. What is the purpose of the window mechanism in TCP?

The purpose of window mechanism in TCP is to indicate the size of the buffer between the sending host and receiving host.

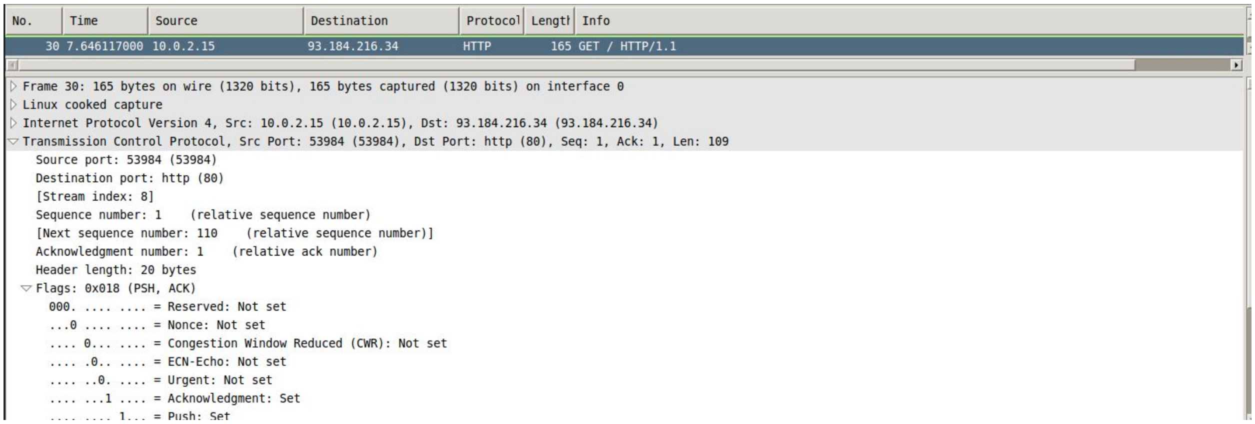
3. What is an MTU? What happens when a packet is larger than the MTU?

MTU is Maximum Transmission Unit, if a packet is larger than the MTU, the packet will be fragmented and reassembled, it will incur a TCP/IP overhead twice.

4. Show (with a Wireshark screenshot) a packet containing a TCP segment, which is piggybacking an ACK



```
 11 4.629215000 10.0.2.15       128.114.142.6   DNS     73 Standard query 0xc908  A example.com
 12 4.629354000 10.0.2.15       128.114.142.6   DNS     73 Standard query 0xf92e  AAAA example.com
 13 4.632786000 128.114.142.6   10.0.2.15       DNS     89 Standard query response 0xc908  A 93.184.216.34
 14 4.635132000 128.114.142.6   10.0.2.15       DNS    101 Standard query response 0xf92e  AAAA 2606:2800:220:1:248:1893:25c8:1946
 15 4.637714000 10.0.2.15       93.184.216.34   TCP     76 51732 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=275909 TSecr=0 WS=128
 16 4.688444000 93.184.216.34   10.0.2.15       TCP     62 http > 51732 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
 17 4.688492000 10.0.2.15       93.184.216.34   TCP     56 51732 > http [ACK] Seq=1 Ack=1 Win=29200 Len=0
 18 4.689305000 10.0.2.15       93.184.216.34   HTTP   165 GET / HTTP/1.1
 19 4.692530000 93.184.216.34   10.0.2.15       TCP     62 http > 51732 [ACK] Seq=1 Ack=110 Win=65535 Len=0
 20 4.706643000 93.184.216.34   10.0.2.15       TCP   1375 [TCP segment of a reassembled PDU]
 21 4.706675000 10.0.2.15       93.184.216.34   TCP     56 51732 > http [ACK] Seq=110 Ack=1320 Win=31656 Len=0
 22 4.706755000 93.184.216.34   10.0.2.15       HTTP   334 HTTP/1.1 200 OK  (text/html)
 23 4.706761000 10.0.2.15       93.184.216.34   TCP     56 51732 > http [ACK] Seq=110 Ack=1598 Win=34294 Len=0
 24 4.708952000 10.0.2.15       93.184.216.34   TCP     56 51732 > http [FIN, ACK] Seq=110 Ack=1598 Win=34294 Len=0
 25 4.709156000 93.184.216.34   10.0.2.15       TCP     62 http > 51732 [ACK] Seq=1598 Ack=111 Win=65535 Len=0
 26 4.733619000 93.184.216.34   10.0.2.15       TCP     62 http > 51732 [FIN, ACK] Seq=1598 Ack=111 Win=65535 Len=0
 27 4.733649000 10.0.2.15       93.184.216.34   TCP     56 51732 > http [ACK] Seq=111 Ack=1599 Win=34294 Len=0
```



```
No.    Time         Source          Destination     Protocol Length Info
    30 7.646117000  10.0.2.15       93.184.216.34   HTTP        165 GET / HTTP/1.1

▷ Frame 30: 165 bytes on wire (1320 bits), 165 bytes captured (1320 bits) on interface 0
▷ Linux cooked capture
▷ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 93.184.216.34 (93.184.216.34)
▽ Transmission Control Protocol, Src Port: 53984 (53984), Dst Port: http (80), Seq: 1, Ack: 1, Len: 109
    Source port: 53984 (53984)
    Destination port: http (80)
    [Stream index: 8]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 110    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Header length: 20 bytes
  ▽ 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
```