Q1:

* + 1. While TCP/IP/ETHERNET is a widely used protocol in practice, ISO OSI model has not found a wide practical application (but possibly is still in use of some legacy systems as it was mentioned in the lecture).
    2. TCP/IP model has 4 layers (Physical, Data Link, Network, Transportation, Application) while ISO OSI model has 7 layers (Physical, Data Link, Network, Transportation, Session, Presentation, Application).
    3. TCP/IP model implements a client-server approach while ISO OSI does not.
    4. OSI’s Transport Layer is connection Oriented while TCP/IP’s Transport Layer is both connection oriented and connectionless.
    5. OSI’s Network Layer provides both connection and connectionless service while TCP/IP’s Network Layer is connectionless.

Q2a:

1. CIDR just like subnetting allows the network to be divided hierarchically. While subnetting does so at byte increments, CIDR divides the address at bit increments:
   1. Subnetting examples:
      1. 8-bit mask: 50.60.70.80/8 = 50.X.X.X (network address) + X.60.70.80 (host address)
      2. 16-bit mask: 150.160.170.180/16 = 150.160.X.X (network address) + X.X.170.180 (host address);
      3. 24-bit mask: 192.168.50.50/24 = 192.168.50.X (network address) + X.X.X.50 (host address);
   2. CIDR examples (eliminates classful addressing):
      1. 255.255.255.255/14 = 255.252.0.0 (current network address) + 0.0.3.255 (the “host” address corresponding to the next 1 of 1024 hosts/subnets in hierarchy) = 1111\_1111.1111\_1100.X.X ORed with X.0000\_0011.1111\_1111.1111\_1111

Q2b:

1. When routing, CIDR allows to vary the length of the prefixes per any level of the backbone internet.
   1. For example, a given network zone can be allocated with a 133.128.0.0/10 (133.1000\_0000.0.0) block of addresses. If one of the next subzones (subtree, LANs) require only 2 ip addresses, we could provided it with the next block 133.128.56.242/31 (133.128.56.1111\_0010). Any packets passing the **XNOR** operation of the IP OF THE PACKET with 133.128.56.1111\_001X would be forwarded out to that zone (this is known as route aggregation where a single routing mask is used for 2 or more addresses instead of having 2 or more entries for each address).

Q2c:

1. CIDR performs IP Routing based on the address prefix of any length which offers next benefit over subnetting when it comes to routing. Let’s say these addresses can be reached over a single port on a router:
2. 123.123.204.0 (0111\_1011.0111\_1011.1100\_1100.0)
3. 123.123.205.0 (0111\_1011.0111\_1011.1101\_1100.0)
4. 123.123.206.0 (0111\_1011.0111\_1011.1110\_1100.0)
5. 123.123.207.0 (0111\_1011.0111\_1011.1111\_1100.0)
6. In the above example, using conventional subnetting, a router would place all 3 addresses into a routing table (or it could use the first 2 octets 123.123. to forward all traffic starting with 123.123.X.X which degrade the bandwidth with the packets that should’ve not been forwarded out of that port). However, using CIDR the 0111\_1011.0111\_1011.1111\_1111.1111\_1111 bitmask can be used in routing table to filter only the packets whose first 22 bits of destination address match 123.123.11XX\_XXXX.XXXX.

Q3:

1. A Private IP Address is an IPv4 address used within you local area network and cannot be used on the internet backbone since it was not assigned to you by IANA. The Private IP Addresses usually live behind a NAT thus translating the internal hosts’ IP Addresses to a single publicly known IP address. In some cases, the port numbers are also translated to some other port number to distinguish the connections between two internal hosts trying to connect to a server with the same ip address and the server’s process with the same port number.
2. These Private IP Address ranges are to be used:
   * 1. 10.X.X.X (Class A)
     2. 172.16.0.0 – 172.31.255.255 (Class B)
     3. 169.254.0.1 – 169.254.255.254 autoconfigured (Class C)
     4. 192.168.X.X (Class C)

Q4:

1. Netwrok Address Translation allows a Router to server as a firewall between an internal private network and the internet backbone (or ISP which in turn can be an internal network of its own between the customer and internet backbone).
2. Using IP and Port translation , a NAT Router allows us to conserve IP Addresses by using an address assigned to it by ISP on its external port and using private IP Addresses on its internal “ports”.
3. A NAT Router intercepts the connections initiated by the hosts on its internal network, replaces internal host’s Source IP Address and if required Source Port number with its own IP Address and Port numbers respectively. It then forwards the connections the modified packets to the outside world. It’s important to mention that NAT Routers only forward packets and do NOT create connections. Using NAT Forwarding table, the NAT Router keeps track of this connections to be able to perform reverse translation for the packets coming from outside world. NAT Forwarding tables includes this information per a given connection:
   * 1. Remote host IP address.
     2. Remote host’s port number
     3. The outside source port number used by the NAT router.
     4. The IP Address of the inside host.
     5. The Port number of the inside host.

Q5:

* + - * 1. First, the DNS is queried for the ip address of the [www.ox.ac.edu](http://www.ox.ac.edu).
        2. The query response provides two IP addresses which are 129.67.242.154 and 129.67.242.155.
        3. Next, the Firefox Browser requests a TCP connection to be established by sending a TCP SYN packet with the fields set to these values:

IP Version: 4 (IPv4)

IP Header Length 20 bytes

IP Total Length: 60

IP Flags: 0x02 (Don’t fragment the packet)

IP Time to Live: 64 hops until the packet is dropped in route.

IP Protocol: TCP protocol is used in Transport Layer of this packet.

IP Source: 192.168.0.19 (by laptop)

IP Destination: 129.67.242.154 (one of the Oxford servers as provided by DNS Response above)

TCP Source Port number: 52314

TCP Destination port: 80 (http)

TCP Sequence Number 2909652809

TCP Header Length: 40 bytes

TCP Flags: 0x002 (SYN is set to establish the connection)

TCP Window Size Value: 29200

TCP Maximum Segment size: 1460 (1500 minus the Header Length of 40 bytes)

TCP Windows Scale: shift count of 7 (left shift by 7 meaning multiply by 2^7=128)

* + - * 1. Next, the Oxford’s server (129.67.242.154) responds with the [SYN,ACK] TCP packet to acknowledge established connection with fields set to these values:

IP Version: 4 (IPv4)

IP Header Length 20 bytes

IP Total Length: 60

IP Flags: 0x02 (Don’t fragment the packet)

IP Time to Live: 44 hops until the packet is dropped in route.

IP Protocol: TCP protocol is used in Transport Layer of this packet.

IP Source: 129.67.242.154 (by laptop)

IP Destination: 129.67.242.154 (one of the Oxford servers as provided by DNS Response above)

TCP Source Port number: 80 (http)

TCP Destination port: 52314

TCP Sequence Number 1202383573

Acknowledgment Number: 2909652810 (ACK+1 = 2909652809 + 1 = 2909652810)

TCP Header Length: 40 bytes

TCP Flags: 0x012 (SYN, ACK are set to acknowledge established connection)

TCP Window Size Value: 28960

TCP Maximum Segment size: 1460 (1500 minus the Header Length of 40 bytes)

TCP Windows Scale: shift count of 7 (left shift by 7 meaning multiply by 2^7=128)

* + - * 1. The firefox browser acknowledges the received [SYN,ACK] above by sending the final ACK:

IP Version: 4 (IPv4)

IP Header Length 20 bytes

IP Total Length: 52

IP Flags: 0x02 (Don’t fragment the packet)

IP Time to Live: 64 hops until the packet is dropped in route.

IP Protocol: TCP protocol is used in Transport Layer of this packet.

IP Source: 192.168.0.19 (by laptop)

IP Destination: 129.67.242.154 (one of the Oxford servers as provided by DNS Response above)

TCP Source Port number: 52314

TCP Destination port: 80 (http)

TCP Sequence Number 2909652810

TCP Acknowledgement Number: 1202383574

TCP Header Length: 40 bytes

TCP Flags: 0x002 (SYN is set to establish the connection)

TCP Window Size Value: 29200

TCP Maximum Segment size: 1460 (1500 minus the Header Length of 40 bytes)

TCP Windows Scale: shift count of 7 (left shift by 7 meaning multiply by 2^7=128)

* + - * 1. Next, the browser requests webpage data using HTTP’s GET request method by sending a packet with the Hypertext Transfer Protocol Layer residing at Application Layer of the packet:

IP Version: 4 (IPv4)

IP Header Length 20 bytes

IP Total Length: 510

IP Flags: 0x02 (Don’t fragment the packet)

IP Time to Live: 64 hops until the packet is dropped in route.

IP Protocol: TCP protocol is used in Transport Layer of this packet.

IP Source: 192.168.0.19 (by laptop)

IP Destination: 129.67.242.154 (one of the Oxford servers as provided by DNS Response above)

TCP Source Port number: 52314

TCP Destination port: 80 (http)

TCP Sequence Number: 2909652810

TCP Next Sequence Number: 2909653268

TCP Acknowledgement Number: 1202383574

TCP Header Length: 32 bytes

TCP Flags: 0x018 (PSH, ACK push and acknowledge)

TCP Window Size Value: 29312

TCP Windows Scale: shift count of 7 (left shift by 7 meaning multiply by 2^7=128)

HTTP Request Method: GET

HTTP Request Version: HTTP/1.1

HTTP Host: .www.ox.ac.uk\r\n (carriage return, new line)

HTTP User-Agent: Mozilla on a Linux machine.

And other HTTP information.

* + - * 1. Next, the server acknowledges the received request by sending an {ACK] response with the ACK field matching the TCP Next Sequence Number of the last packet (HTTP’s GET request):

IP Version: 4 (IPv4)

IP Header Length 20 bytes

IP Total Length: 52

IP Flags: 0x02 (Don’t fragment the packet)

IP Time to Live: 44 hops until the packet is dropped in route.

IP Protocol: TCP protocol is used in Transport Layer of this packet.

IP Source: 129.67.242.154 (by laptop)

IP Destination: 129.67.242.154 (one of the Oxford servers as provided by DNS Response above)

TCP Source Port number: 80 (http)

TCP Destination port: 52314

TCP Sequence Number 1202383574

Acknowledgment Number: 2909653268

TCP Header Length: 32 bytes

TCP Flags: 0x010 (ACK is set to acknowledge the request)

TCP Window Size Value: 235

TCP Windows Scale: shift count of 7 (left shift by 7 meaning multiply by 2^7=128)

* + - * 1. Next, a series of TCP packets containing the webpage are sent from the server to the firefox browser with each delivered packet an ACK response sent by the firefox browser (kindly see Packets 57 thru 75 in the oxford\_wireshark\_with\_fin.pcapng file).
        2. Six more [SYN] packets are sent by the browser requesting additional website data. It should be noted that each of this packets had a different Seq and Port (52314, 52316, 52318, 52320, 52324, 52326) numbers!
        3. Finally, for each established connection, a [FIN, ACK] is sent out by the server once the browser application is closed and the connections is abrupted by the client.

IP Version: 4 (IPv4)

IP Header Length 20 bytes

IP Total Length: 52

IP Flags: 0x02 (Don’t fragment the packet)

IP Time to Live: 44 hops until the packet is dropped in route.

IP Protocol: TCP protocol is used in Transport Layer of this packet.

IP Source: 129.67.242.154 (one of the Oxford servers as provided by DNS Response above)

IP Destination: 192.168.0.19 (by laptop)

TCP Source Port number: 80 (http)

TCP Destination port: 52314

TCP Sequence Number1202402091

TCP Acknowledgement Number: 2909653905

TCP Header Length: 40 bytes

TCP Flags: 0x011 (FIN, ACK)

TCP Window Size Value: 245

TCP Windows Scale: shift count of 7 (left shift by 7 meaning multiply by 2^7=128)

* + - * 1. For each [FIN, ACK] packet received from the server, one [ACK] response is sent out to the Oxford server with the fields as shown below:

IP Version: 4 (IPv4)

IP Header Length 20 bytes

IP Total Length: 52

IP Flags: 0x02 (Don’t fragment the packet)

IP Time to Live: 64 hops until the packet is dropped in route.

IP Protocol: TCP protocol is used in Transport Layer of this packet.

IP Source: 192.168.0.19 (by laptop)

IP Destination: 129.67.242.154 (one of the Oxford servers as provided by DNS Response above)

TCP Source Port number: 52314

TCP Destination port: 80 (http)

TCP Sequence Number 2909653905

TCP Acknowledgement Number: 1202402092

TCP Header Length: 32 bytes

TCP Flags: 0x010 (ACK)

TCP Window Size Value: 568

TCP Windows Scale: shift count of 7 (left shift by 7 meaning multiply by 2^7=128)

Q6:

1. Performing traceroute ox.ac.uk, next output was received:

[root@z\_book\_vm hw01]# cat traceroute.txt

traceroute to www.ox.ac.uk (129.67.242.154), 30 hops max, 60 byte packets

1 gateway (192.168.0.1) 2.600 ms 2.564 ms 13.332 ms

2 96.120.74.237 (96.120.74.237) 14.241 ms 14.243 ms 14.213 ms

3 68.85.118.205 (68.85.118.205) 15.870 ms 15.859 ms 18.086 ms

4 96.108.50.9 (96.108.50.9) 19.808 ms 19.794 ms 19.777 ms

5 be-101-ar03.plainfield.nj.panjde.comcast.net (96.108.49.105) 24.020 ms 24.006 ms 23.990 ms

6 be-33659-cr01.newark.nj.ibone.comcast.net (68.86.90.21) 28.463 ms 18.401 ms 19.645 ms

7 be-10203-cr02.newyork.ny.ibone.comcast.net (68.86.85.186) 18.568 ms 18.517 ms 20.852 ms

8 be-10368-pe01.111eighthave.ny.ibone.comcast.net (68.86.84.218) 20.809 ms 22.746 ms 23.533 ms

9 nyk-b5-link.telia.net (62.115.148.44) 22.681 ms 23.459 ms 23.429 ms

10 nyk-bb4-link.telia.net (80.91.254.15) 98.410 ms \* 99.808 ms

11 ldn-bb4-link.telia.net (62.115.112.245) 92.689 ms ldn-bb3-link.telia.net (62.115.113.21) 105.472 ms ldn-bb4-link.telia.net (62.115.112.245) 95.583 ms

12 ldn-b7-link.telia.net (62.115.138.155) 95.541 ms 87.278 ms ldn-b7-link.telia.net (62.115.138.151) 92.296 ms

13 jisc-ic-345130-ldn-b7.c.telia.net (62.115.175.107) 94.165 ms 94.146 ms 94.127 ms

14 ae24.londtt-sbr1.ja.net (146.97.35.193) 97.485 ms 94.096 ms 94.071 ms

15 ae28.londtw-sbr2.ja.net (146.97.33.62) 96.215 ms 91.162 ms 103.937 ms

16 ae30.londpg-sbr2.ja.net (146.97.33.5) 105.509 ms 94.733 ms 94.701 ms

17 ae19.readdy-rbr1.ja.net (146.97.37.194) 100.117 ms 96.258 ms 96.586 ms

18 ae2.oxfoii-rbr1.ja.net (193.63.108.94) 94.705 ms 94.692 ms 99.986 ms

19 ae3.oxforq-rbr1.ja.net (193.63.108.98) 98.052 ms 101.336 ms 99.269 ms

20 oxford-university.ja.net (193.63.109.90) 102.034 ms 96.314 ms 96.631 ms

21 \* \* \*

22 \* \* \*

23 oxweb-router.it.ox.ac.uk (129.67.242.158) 98.245 ms 102.172 ms 100.507 ms

1. From the output shown above, it can be seen that:
   1. a total of 23 hops are required to reach the oxford server.
   2. The average round trip time is 98.326 ms.
   3. Making an assumption that ISPs such as comcast can be considered to be a part of internet backbone, the routers between the gateway (hop=1) and the oxford network (starting at hop=18) are all a part of the internet backbone. Otherwise, the router on the residing on the telia.net domain are the internet backbone (hop=9 thru hop=13).