Ans1:

TCP/IP model was a protocol set originally developed by the USA Department of Defense [1] and consists of four layers; Application layer, Transport layer, Network layer, Link layer. Open Systems Interconnection model (OSI Model) on the other hand was developed by ISO (International Organization of Standardization) [2] and consists of 7 layers; Application layer, Presentation layer, Session layer, Transport layer, Network layer, Link layer and Physical layer

The major obvious difference is that OSI Model has 2 more layers than the TCP/IP model, the differences run deeper than that though, OSI model is a conceptual, protocol independent model whereas TCP/IP model is based on standard protocols. OSI Model has a separate Presentation and Session Layer which deals with tasks like data translation and encryption [3]. While in the TCP/IP model these tasks are assumed to be performed by the Application layer because in the end they're happening at the end user on purely software level without anything to do with how the data was transmitted.

Ans2:

In my opinion, a circuit switched network would be more suitable for the given scenario. Packet Switched networks are great when the connection details are unknown and uncertain, by details I refer to the amount of data being sent and for how long, a packet switched network can be great at dealing with the uncertainty even if it can't provide a guaranteed delivery.

While in this case, we already know that N bytes of data will be sent at a steady rate, so bandwidth and a circuit path can be assigned for the application without wasting too much resources. Using circuit switching in this case would also increase the reliability of the application; now that we have guaranteed data delivery.

Ans3:

A. Size = 150 MB
Size can also be represented as = 150 x 8 = 1200 Megabits
Each Switch is using store and forward switching, and all three links have the same bandwidth that is 5Mbps,
Hence the time it takes for the packet to travel from source to destination would be
= (1200 / 5) * 3
= 240 * 3
= 720 seconds

B. Dividing the original data into 10 packets of size = 15MB
Also, can be represented as 15 x 8 = 120Mb
Time for the first packet to travel to the first switch would be
= 120 / 5
= 24 seconds
Time the first packet will take to reach the second switch will be 24 + 24 = 48 seconds.
The second packet will leave for the first switch when the first packet leaves the first switch, it would take a total of 24 seconds to reach the first switch as well.

So total elapsed time when second packet reaches the first switch would be **48 seconds.**

C. Working from my previous answers, we can observe the total elapsed time first packet will take to reach the destination would be = 24*3 = 72 seconds
And then total time second packet will reach would be 24*3 + 24 = 96 seconds
And so, on each next packet would take 24 more seconds to reach the destination, we can derive the following,
24*3 + (n-1)*24 where n is the n^th packet.
For 10 packets, 24*3 + 9*24 = **288 seconds**

D. Overall Transmission delay would not be affected at all if the length of link between the switch is doubled. Transmission delay is dependent upon the bandwidth of the link not the distance, propagation delay is dependent upon the length of the link.

E. Pros

- Transmission can be faster because message segmentation can use pipelining.
- More efficient, as an example if there is some data loss, just a segment has to be sent again rather than the entire message.

Cons

- Each segment and packet would require its own headers and other essential routing information resulting in bandwidth wastage.
- Resource wastage at user level, dividing up into packets and assembling a bunch of packets back into a message takes processing power.

<u>Ans4:</u>

An IP address or Internet Protocol address can be defined as a numerical value assigned to a device or a computer connected to some kind of computer network, the said the numerical value is used to identify a unique device for communication purposes.[4]

A MAC address is a unique identifier unique to every device(or each Network Interface Controller). MAC address can be used to identify different devices on a network, they also help provide more information about the device, such as; Make(Apple, Google..), Device Type(PDA, Laptop..), etc.

Now, in an average network these days we need both IP addresses and a MAC Addresses to communicate with each other. To visualize why this is required let's pretend we're the ones creating a network from scratch. Now we have multiple computers who would like to communicate with each other, now to do that they need to know who they're talking to, so we assign the computers a unique ID. Now using those IDs, the computers can tell who they're talking to but how does one computer know how to relay it's data to another one? We can imagine the unique IDs as an SSN, now SSNs help us realize who we're talking to but it's not of much help when it comes to tracking down the path to the person with a certain SSN. Now, this is where IP Addresses come in, they work as our current address, IP addresses change as we move places in a similar fashion addresses in real-life work.

As an example, I am currently at the snell library, and there is a piece of mail that has to be urgently delivered to me, my NUID alone wouldn't really be of much help to track me down, neither would be my current address alone(the mailman wouldn't know how to distinguish me), but if he had my NUID and my address, the mailman could walk into snell start to shout my NUID, I'd probably listen and accept the mail right? And that's pretty much how MAC Addresses and IP addresses work. Our networks IP Address would get the packets to our router and then our router would check the local IP address and MAC Addresses to decide to which computer the packet is meant for.

<u>Ans5:</u>

```
import math
print("Enter a number")
inputvar = int(input())
counter = 0
i = 0

if inputvar > 1:
    for i in range(2,int(math.sqrt(inputvar))+1):
        if (inputvar % i) == 0:
            print("Not a Prime")
            break
    else:
        print("Is a Prime")
else:
    print("Numbers equal to 1 or smaller cannot be prime")
```

Output:

```
~ » python3 primenumber.py
Enter a number
21
Not a Prime
~ » _
```

Ans6:

```
import string

print("Input a string:")

strinput = input()
strinput = strinput.replace(' ','').lower()
strrev = strinput[::-1]

print("String you input is: ",strinput)
print("String Reversed is: ",strrev)

if strinput == strrev:
      print("Hence string is a palindrome")
else:
      print("Hence string ain't a palindrome")
```

Output:

```
~ » python3 palindrome.py
Input a string:
step on no pets
String you input is:  steponnopets
String Reversed is:  steponnopets
Hence string is a palindrome
```

Ans7:

A. Distance between the two hosts = 15,000 km = 15,000,000 meter
Speed over link = 2.2 x 10^8 meters/sec

Hence, propagation delay over the link is 15,000,000 / 220,000,000 = 0.0681 seconds
Bandwidth delay product is known as product of the links bandwidth and its round-trip delay time.

The round-trip delay time is = 0.0681 * 2 = 0.1382 seconds

Hence the propagation delay time product is 0.1382 * 1.5Mbps
= 0.2073 Mb
= 207.3 Kb or 207300 Bits
= 25.9125 KB

B. Bandwidth delay product is the maximum number of bits that can be in the link at any given time, hence the answer is 207300 bits.

C. Bandwidth delay product is the product of the bandwidth of the link and the round-trip delay time. Round trip delay time is total amount of time a message takes to reach its destination + the amount of time its acknowledgement takes to reach back. Hence Bandwidth delay can be described as the maximum amount of data that can be in the link at any given time.

D. Width of the link would be 15000 km / 207300
= 15000 * 1000 / 207300
= 72.358m

Maximum width of a football (soccer, not the American eggball) field is 90m, so no, the width of the link is not bigger than a football fields width.

E. Width of the link can be written as = Length of the link / Bandwidth-Delay Product
= length / (2*bandwidth * length /speed)
= (length * speed) / (2 * length * bandwidth)
= time / 2*bandwidth

Hence, = s/2R

Referenes

[1] Cerf, Vinton G. & Cain, Edward (1983), "The DoD Internet Architecture Model", Computer Networks, 7, North-Holland, pp. 307–318

[2] "OSI The Internet That Wasn't". IEEE Spectrum. March 2017.

[3] Dean, Tamara (2010). *Network+ Guide to Networks*. Delmar. pp. 44–47.

[4] Frystyk, Henrik (1994). "The Internet Protocol Stack"