Solihull College

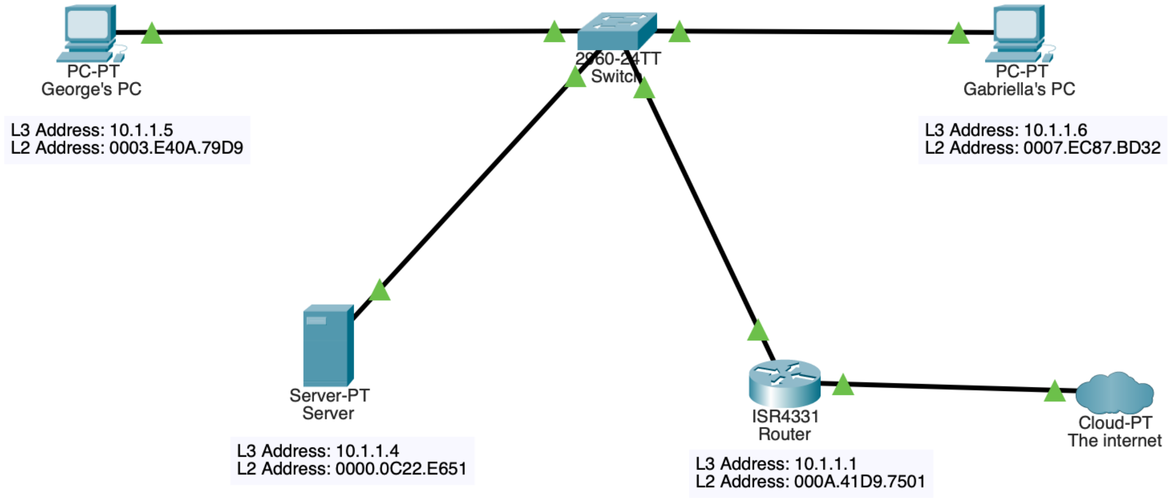
Assignment 1

Unit 10

George Hotten

How Data Travels Around a Network

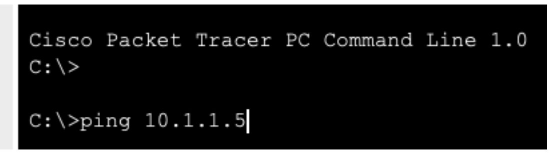
There are many steps to data traveling around a network. For this example, I shall be using the following network I created in Packet Tracer:



I shall present each step with separate headings including which layer it falls under on the OSI model.

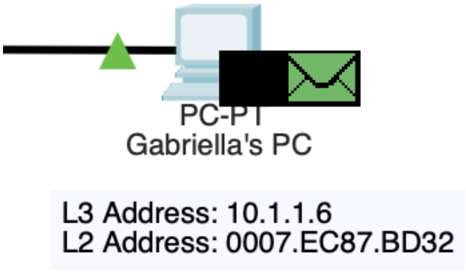
Step 1, Layer 7

Here the application (such as the Command Line running the ping command) has data it wants to send across the network. The application begins to create the request and adds application-specific headers and footers to the request.



```
Cisco Packet Tracer PC Command Line 1.0
C:\>

C:\>ping 10.1.1.5
```

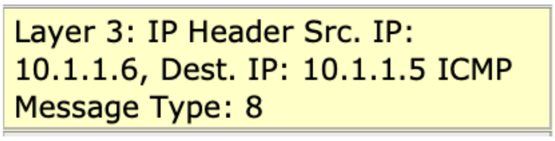


Step 2, Layer 4

Here the data begins its encapsulation. At this step, the application data is encapsulated into a segment which contains the protocol used to send the data. Either TCP or UDP. The destination ports are also added on this layer (for example, a web request would use the port 80).

Step 3, Layer 3

Here the data is further encapsulated into a packet which contains the source and destination IP of where the data is being sent to/from.



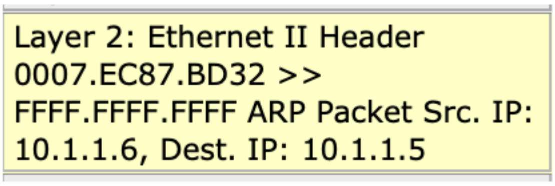
Layer 3: IP Header Src. IP: 10.1.1.6, Dest. IP: 10.1.1.5 ICMP Message Type: 8

Step 4, Layer 2

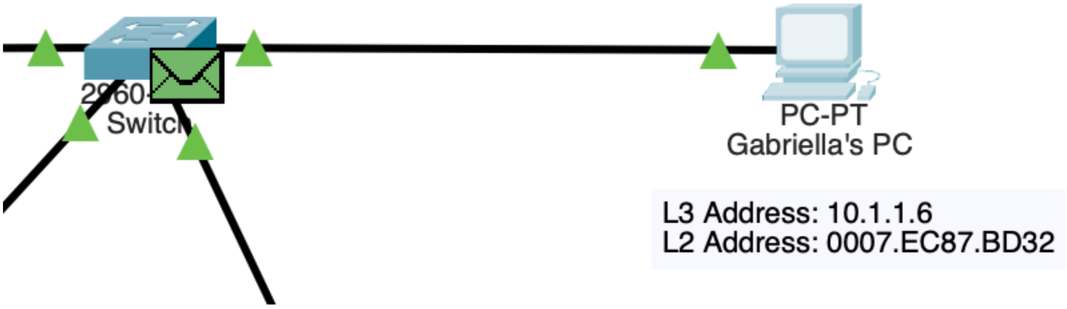
Once again, the data is further encapsulated into a frame which contains the source and destination MAC address where the data is being sent to or from. However, it is highly likely that the source computer does not know the MAC address of the destination computer. To resolve this, the computer sends out an Address Resolution Protocol (ARP) request.

Step 4.1, Layer 2 & 1

The computer creates a frame where the destination MAC address is FFFF.FFFF.FFFF.FFFF, which is a broadcast address (meaning the frame should be sent to every device on the network). This is then sent across the physical wires at Layer 1 to the switch.



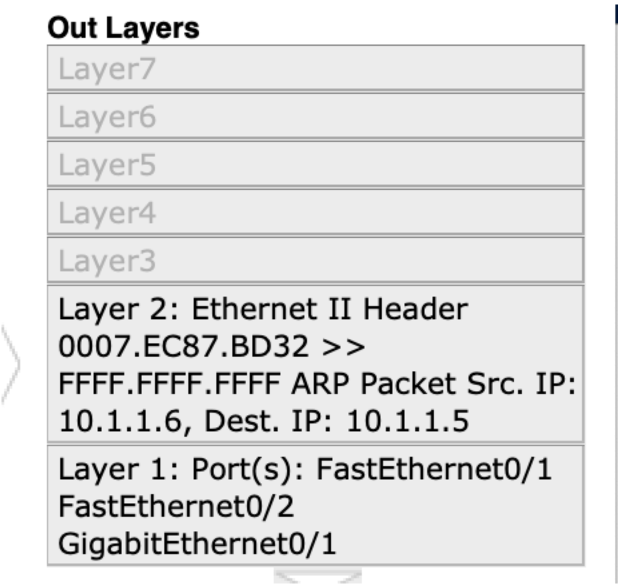
Layer 2: Ethernet II Header 0007.EC87.BD32 >> FFFF.FFFF.FFFF ARP Packet Src. IP: 10.1.1.6, Dest. IP: 10.1.1.5



L3 Address: 10.1.1.6
L2 Address: 0007.EC87.BD32

Step 4.2, Layer 2 & 1

Once the switch receives the frame, it de-encapsulates it to read the MAC address. When it identifies it is the broadcast address, the switch encapsulates the frame and sends it out to every connected device.



Out Layers

Layer7

Layer6

Layer5

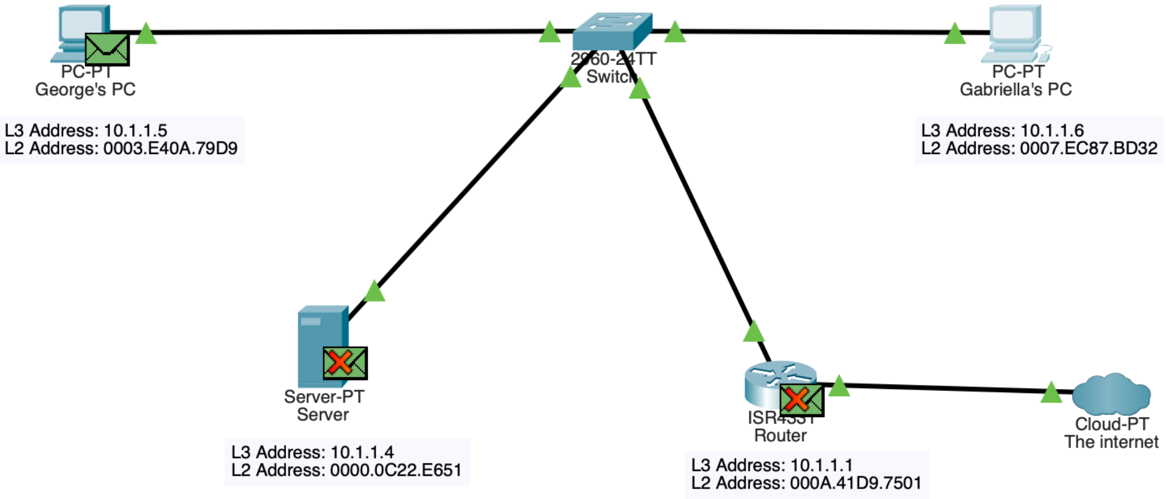
Layer4

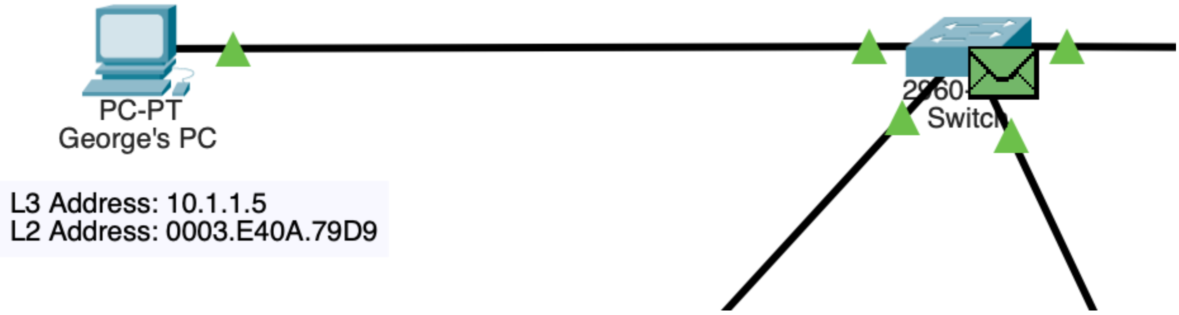
Layer3

Layer 2: Ethernet II Header 0007.EC87.BD32 >> FFFF.FFFF.FFFF ARP Packet Src. IP: 10.1.1.6, Dest. IP: 10.1.1.5

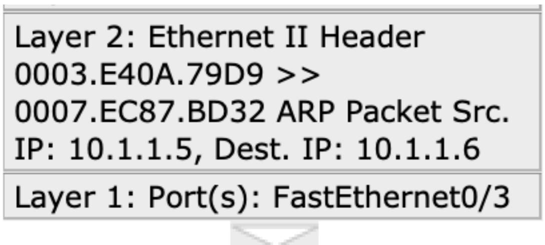
Layer 1: Port(s): FastEthernet0/1 FastEthernet0/2 GigabitEthernet0/1

Once the end devices receive the ARP request, they de-encapsulate it to see if the requested IP Address belongs to them. If it does, they add their MAC address, encapsulate the frame, and send it back. Once the switch receives the frame, it de-encapsulates it, finds the destination MAC address, encapsulates it, and sends it to the requested device.



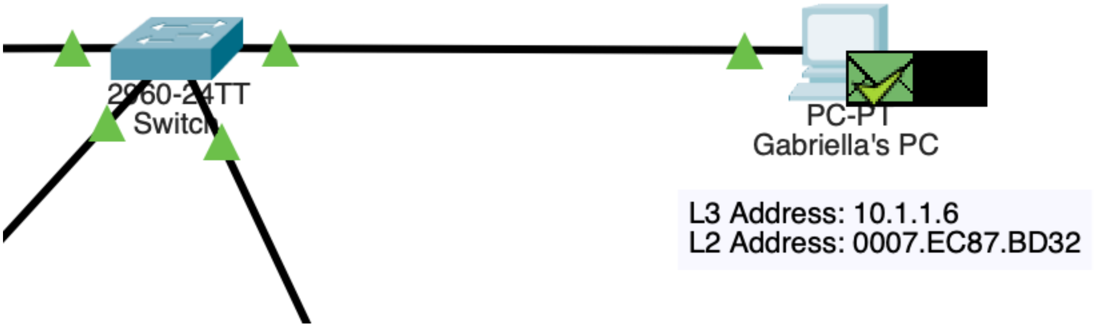


If the IP address does not match, the frame is ignored.



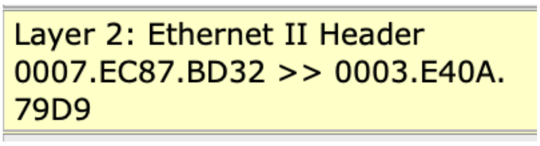
Layer 2: Ethernet II Header 0003.E40A.79D9 >> 0007.EC87.BD32 ARP Packet Src. IP: 10.1.1.5, Dest. IP: 10.1.1.6

Layer 1: Port(s): FastEthernet0/3

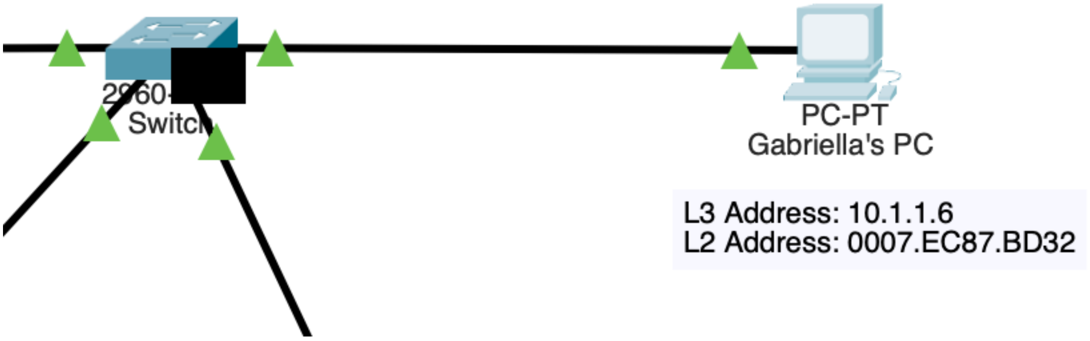


Step 5, Layer 2 & 1

Now the source computer has the desired MAC Address, it adds it to the Layer 2 header and encapsulates the frame. The frame is then sent across the ethernet wires in binary via the device's Network Interface Card.

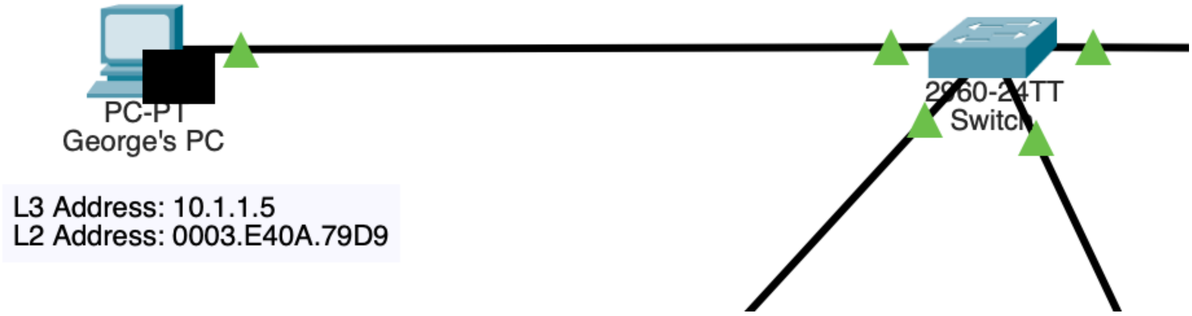


Layer 2: Ethernet II Header 0007.EC87.BD32 >> 0003.E40A.79D9



Step 6, Layer 2 & 1

The switch receives the frame and de-encapsulates the frame to read the destination MAC Address. When the destination machine sent back its ARP response, the switch logged its MAC Address in its address table, so it knows exactly which port to send the data down. The switch then encapsulates the frame and sends it to the desired destination.



Step 7, Layer 1

The destination device receives the packet.

Step 8, Layer 2

The destination device de-encapsulates the frame and reads its L2 data to check the MAC address. If the destination MAC address doesn't match its own, it disregards the frame.

Step 9, Layer 3

As the MAC address matched, the device further de-encapsulates the frame into a packet to read the L3 data to check the IP address. If the destination IP address doesn't match its own, it disregards the packet.

Step 10, Layer 4

As both the IP and MAC address match, the device knows the data is intended for it. It then further de-encapsulates the packet to read the L4 information and forwards the segment to the specified application service.

Step 11, Layer 7

The application has now received the segment and de-encapsulates it to get the original data that was sent from the source. The application can now read and respond to that data. For example, sending a website back to the source.

Protocols & Standards

Wi-Fi

Wi-Fi is the technology used to connect millions of devices to the internet via the transmissions of radio waves back and forth between a router. The said router then uses a wired connection to connect to the internet. Wi-Fi usually operates on 2 frequencies: 2.4GHz or 5GHz.

Transmission Control Protocol/Internet Protocol (TCP/IP)

This is a set of protocols used to connect network devices on the internet.

Domain Name Server (DNS)

A DNS is a server that takes a URL such as google.com and turns it into an IP that the computer can then use to send web requests to that server.

Dynamic Host Configuration Protocol (DHCP)

DHCP is a protocol that assigns IP addresses to nodes in a network to allow it to connect to other devices. This is often seen in a client-server network configuration.

Hyper Text Transfer Protocol (HTTP)

HTTP is the protocol used when requesting resources from a web server, for example a html file or pictures. This is used in a client-server configuration as all requests are initiated from the client and the server responds to them.

File Transfer Protocol (FTP)

FTP is the protocol used to send and receive files between two devices over a network and/or the internet.

Simple Mail Transfer Protocol (SMTP)

SMTP is the protocol used to send and receive emails between two or more mail servers. This is then retrieved using either the Post Office Protocol or the Internet Message Access Protocol.

IrDA

IrDA, aka the Infrared Data Association, is a group that provides a suite of wireless ‘line-of-sight’ protocols that provides connectivity between devices. This can often be found in devices such as TV remotes.

3G, 4G and 5G

3G, 4G and 5G are technology used to connect millions of ‘cellular’ enabled devices across the world – usually phones or tablets. They work through ‘cellular’ technology where signals are passed from phone tower to phone tower until it reaches the recipient. With every generation, the speed and bandwidth increases usually by larger amounts. However, this comes at the cost of using shorter waves which can hinder connectivity.

Wireless Application Protocol

Wireless Application Protocol (WAP) is a set of protocols used by wireless devices such as phones or radio transceivers that standardizes how they can be used for internet access. WAP especially optimizes internet access for mobile devices such as by using the Wireless Markup Language to deliver web pages.

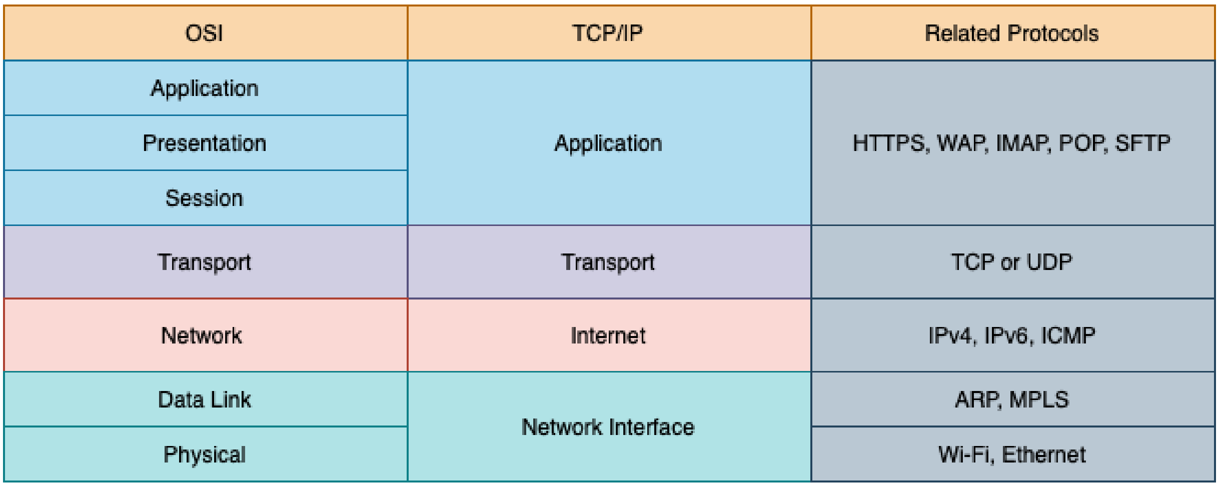
WPA and WPA2

WPA, aka Wi-Fi Protected Access, is the protocol used to secure Wi-Fi network by using 256-bit keys for encryption. This is done via the Temporal Key Integrity Protocol which generates a new key for each packet of data. However, once WPA2 came along, TKIP was dropped in favour of AES which provides a stronger encryption.

Why are network standards and protocols needed?

Network standards and protocols are needed so devices all around the world can connect as they all use the same methods of transmission. This means everything is compatible with everything and data can be sent around the world easily. If the standards and protocols were different everywhere, it would make the internet a lot harder to maintain and it would take a lot of work to support all the different standards.

The OSI & TCP/IP Model and their protocols

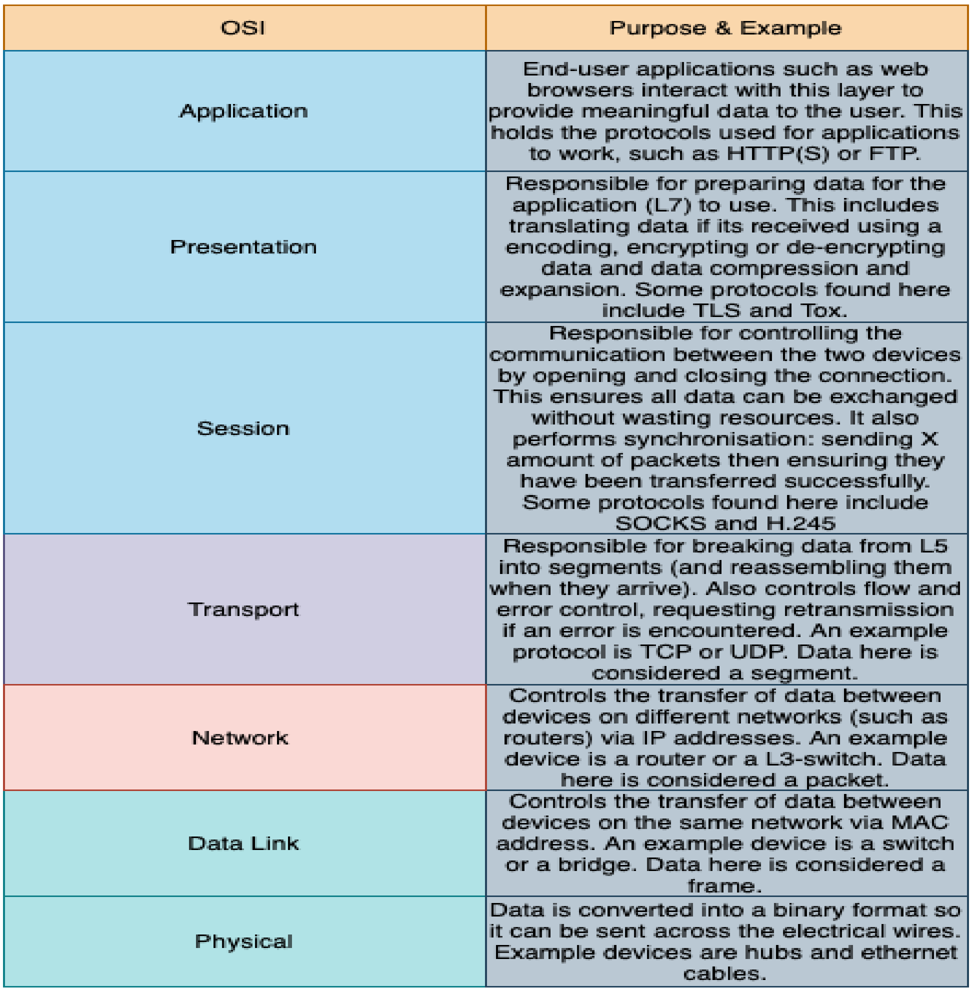


OSI	TCP/IP	Related Protocols
Application	Application	HTTPS, WAP, IMAP, POP, SFTP
Presentation		
Session		
Transport	Transport	TCP or UDP
Network	Internet	IPv4, IPv6, ICMP
Data Link	Network Interface	ARP, MPLS
Physical		Wi-Fi, Ethernet

OSI vs. TCP/IP Model

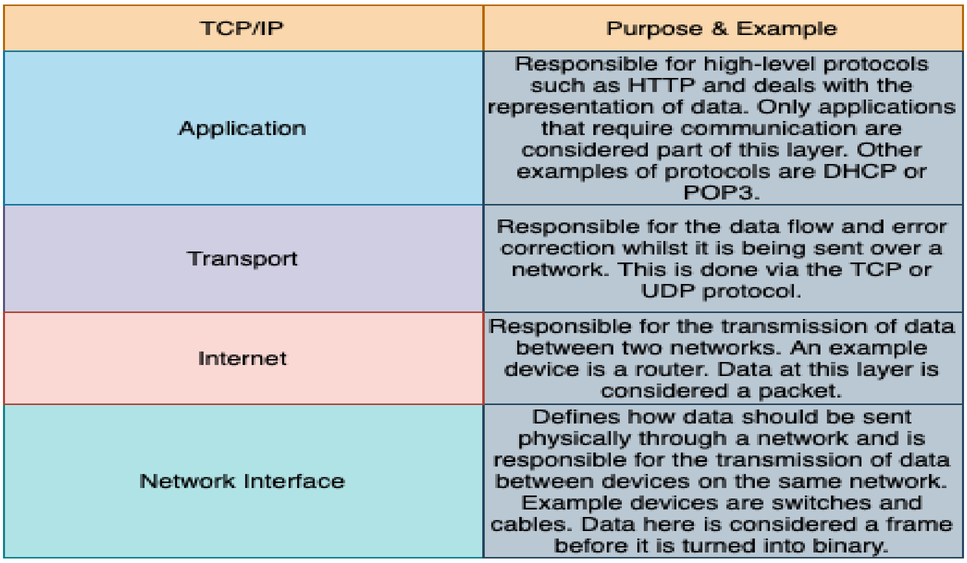
The OSI and TCP/IP models are used to represent how data travels around a network. The OSI model is considered more extensive as it has more layers, 7 compared to 5, whilst the TCP/IP model hides some of the complexity of lowest and highest layers.

Purpose of Each Layer: OSI Model



OSI	Purpose & Example
Application	End-user applications such as web browsers interact with this layer to provide meaningful data to the user. This holds the protocols used for applications to work, such as HTTP(S) or FTP.
Presentation	Responsible for preparing data for the application (L7) to use. This includes translating data if its received using a encoding, encrypting or de-encrypting data and data compression and expansion. Some protocols found here include TLS and Tox.
Session	Responsible for controlling the communication between the two devices by opening and closing the connection. This ensures all data can be exchanged without wasting resources. It also performs synchronisation: sending X amount of packets then ensuring they have been transferred successfully. Some protocols found here include SOCKS and H.245
Transport	Responsible for breaking data from L5 into segments (and reassembling them when they arrive). Also controls flow and error control, requesting retransmission if an error is encountered. An example protocol is TCP or UDP. Data here is considered a segment.
Network	Controls the transfer of data between devices on different networks (such as routers) via IP addresses. An example device is a router or a L3-switch. Data here is considered a packet.
Data Link	Controls the transfer of data between devices on the same network via MAC address. An example device is a switch or a bridge. Data here is considered a frame.
Physical	Data is converted into a binary format so it can be sent across the electrical wires. Example devices are hubs and ethernet cables.

The Purpose of Each Layer: TCP/IP Model



TCP/IP	Purpose & Example
Application	Responsible for high-level protocols such as HTTP and deals with the representation of data. Only applications that require communication are considered part of this layer. Other examples of protocols are DHCP or POP3.
Transport	Responsible for the data flow and error correction whilst it is being sent over a network. This is done via the TCP or UDP protocol.
Internet	Responsible for the transmission of data between two networks. An example device is a router. Data at this layer is considered a packet.
Network Interface	Defines how data should be sent physically through a network and is responsible for the transmission of data between devices on the same network. Example devices are switches and cables. Data here is considered a frame before it is turned into binary.

What is the difference?

As seen above, the OSI and TCP/IP are very similar in-terms of what each layer does. The main difference between the two models is that the TCP/IP model has 4 layers compared to the OSI's 7. This is because the OSI is considered a reference model for how networks are built, whereas the TCP/IP model is used to show how devices connect to the internet.

TCP/IP combines layers 1 and 2 (physical and data link) into the Network Interface layer. It also combines layer 5, 6 and 7 (session, presentation, and application) into the Application layer. This is because the OSI focuses more on the standardization of devices such as switches, routers and NICs, whereas TCP/IP does not standardize any hardware and focuses on the actual connection of devices.

Conclusion

Personally, I am in favour of the OSI model as it helps me understand how data is prepared and received in a higher level of detail, instead of clumping parts together as in the TCP/IP model. It also allows for a deeper knowledge of how connections are established and maintained and how data is encrypted and compressed to speed up transmission and keep it secure.

DTE vs. DCE

What is Data Terminal Equipment (DTE)?

DTE are often considered end devices that receive binary digital data. These devices can both send and receive digital data. Some examples are printers, computers, and faxes.

What is Data Circuit Equipment (DCE)?

A DCE is a device that can transmit or receive data in an analog or digital signal. A DCE takes the signal and converts it into the correct format. For example, when data is being received from a telephone line it will be Analog and a DCE converts that to digital for the router to understand. An example of a device that does this a modem, sometimes a router if they are a router-modem combination.

Identifying DTE, DCE or neither

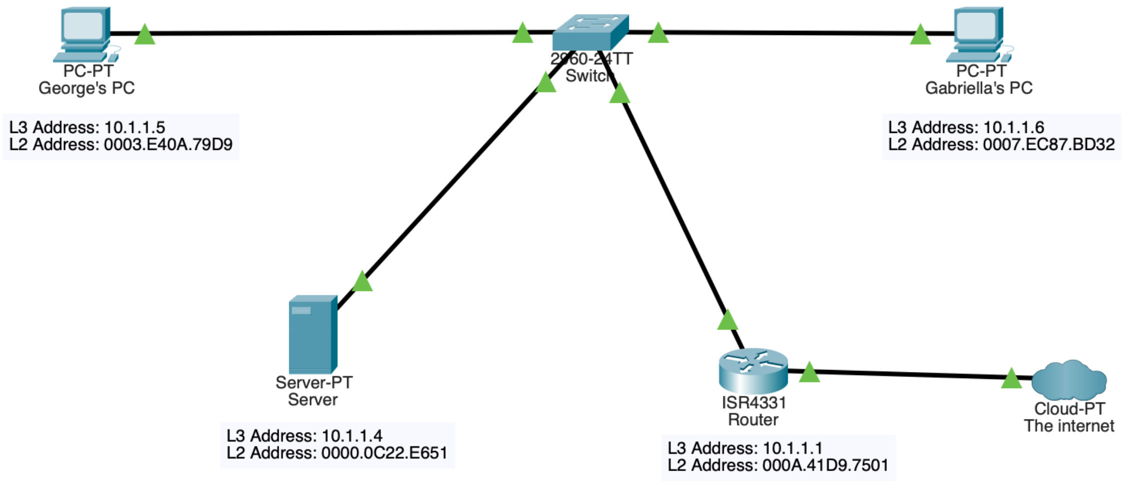
1. A device that works at layer 3 of the OSI model and can communicate between networks using IP addresses – **DTE – Router**
2. A layer 1 device that broadcasts all signals it receives – **DCE – Hub**
3. An end user device that can be connected to a network wired or wirelessly and can be carried around – **DTE – Laptop**
4. A handheld device that is used for voice and data using 4G, 5G and wireless networks – **DTE – Mobile Phone**
5. A network device that converts digital signals to analogue and back – **DCE – Modem**
6. An end user device that can be connected to a network and usually sits on a desk – **DTE – Desktop PC**
7. A component that allows a device to connect to a network through a WAP – **DCE – Wireless NIC**

Data Elements

Addresses

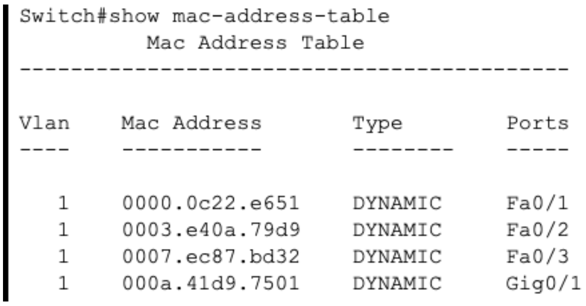
MAC addresses are used to identify specific devices on a network. Each MAC address is unique to the Network Interface Card found inside it. IP Addresses are also used to identify devices, mainly routers, when communicating on the internet. However, IP addresses can also be found within a network under private subnets. IP addresses found on the internet must be unique yet can be changed.

Let's look at this in our example from Task 1.



As you can see from the comments, each of our devices have an IP address (L3 Address) and a MAC address (L2 Address). The IPs have been configured manually under the private Class A IPs. The MAC addresses were assigned when they were created and cannot be changed as they are burnt into the card.

MAC Addresses are used at Layer 2 at the OSI model, meaning they are handled by the switch. The switch logs which MAC address is at what port when data is sent around the network it can be sent to the correct device.

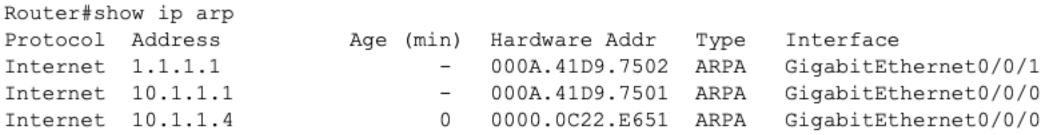


```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0000.0c22.e651    DYNAMIC     Fa0/1
   1    0003.e40a.79d9    DYNAMIC     Fa0/2
   1    0007.ec87.bd32    DYNAMIC     Fa0/3
   1    000a.41d9.7501    DYNAMIC     Gig0/1
```

IP Addresses are used at Layer 3 of the OSI model, meaning they are handled by the router. The router logs what IP address correlates to which MAC address to confirm the data being sent into the network is in the right place.



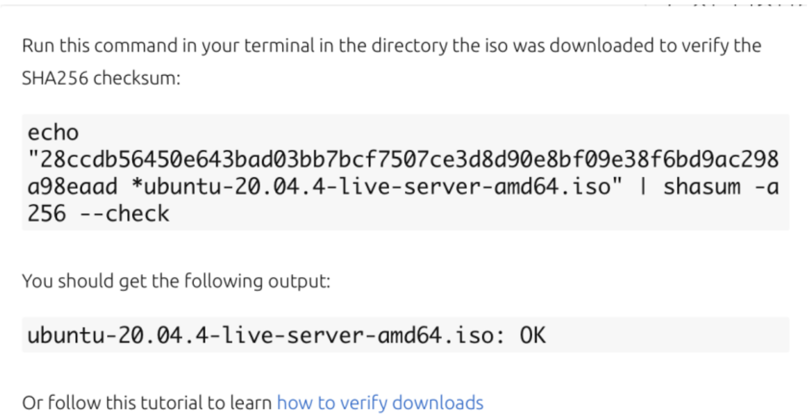
```
Router#show ip arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  1.1.1.1                 -   000A.41D9.7502  ARPA   GigabitEthernet0/0/1
Internet  10.1.1.1                -   000A.41D9.7501  ARPA   GigabitEthernet0/0/0
Internet  10.1.1.4                0   0000.0C22.E651  ARPA   GigabitEthernet0/0/0
```

Addressing is important as it allows for data to be sent around a network with intended recipients. If addressing didn't exist, data would not be able to reach other people.

Checksum

A checksum is a method of verifying the integrity of data. Using hashing algorithms, a file can be hashed into a string of letters and numbers. This is especially useful when sending files over the internet. Websites can provide a checksum along with their download and the user can then verify the checksum by re-hashing the file and see if the checksums match. If they don't the file is different to the one you intended to download.

As an example, I have downloaded the Ubuntu server ISO. To ensure the file has downloaded as intended, I ran the following commands:



Run this command in your terminal in the directory the iso was downloaded to verify the SHA256 checksum:

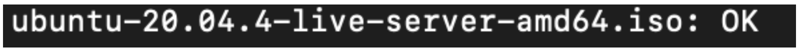
```
echo
"28ccdb56450e643bad03bb7bcf7507ce3d8d90e8bf09e38f6bd9ac298
a98eaad *ubuntu-20.04.4-live-server-amd64.iso" | shasum -a
256 --check
```

You should get the following output:

```
ubuntu-20.04.4-live-server-amd64.iso: OK
```

Or follow this tutorial to learn how to verify downloads

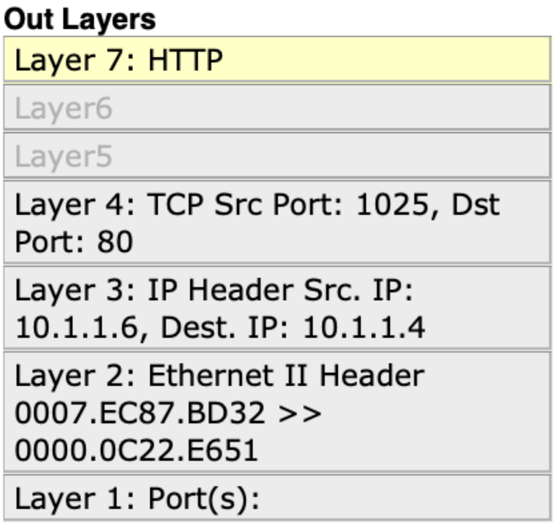
After running the command, I received the following output meaning my file downloaded correctly.



```
ubuntu-20.04.4-live-server-amd64.iso: OK
```

Encapsulation

Encapsulation is the method of adding extra information into data travelling through the layers of the OSI or TCP/IP model. For example, Layer 3 on the OSI encapsulates the data adding the source and destination IP address.



Out Layers

Layer 7: HTTP

Layer6

Layer5

Layer 4: TCP Src Port: 1025, Dst Port: 80

Layer 3: IP Header Src. IP: 10.1.1.6, Dest. IP: 10.1.1.4

Layer 2: Ethernet II Header 0007.EC87.BD32 >> 0000.0C22.E651

Layer 1: Port(s):

As you can see, data is encapsulated at each layer adding some extra information as it goes down the OSI stack.

This is important for ensuring that data can travel around the network smoothly, providing the correct data so it can reach the correct recipient.

Sequence Numbers

Sequence numbers are part of TCP transmission to help keep track of data to ensure all data is arriving. Each segment contains a sequence number. When the segment is first created, it starts at a sequence number of 0. Every time data is sent, the sequence number is incremented by the size of the frame. For example, the sequence number for a segment is X, and the length of the segment is Y. If the segment arrives successfully, then the sequence number for the next segment should be X+Y.

When data is received, the sequence number is checked. If the sequence number isn’t what was expected, the data is re-requested and re-transmitted. When the data is received, the TCP protocol can re-arrange the data to the correct order.