Assignment No 01

Have you ever wondered why you need an IP address to access the Internet? Is it true that an IP address can uniquely identify the user? Are you curious to learn what the life of a packet looks like? If the answer is yes, let’s dive in!

This room is the first room in a series of four rooms dedicated to introducing the user to vital networking concepts and the most common networking protocols:

* Networking Concepts (this room)
* [Networking Essentials](https://tryhackme.com/r/room/networkingessentials)
* [Networking Core Protocols](https://tryhackme.com/r/room/networkingcoreprotocols)
* [Networking Secure Protocols](https://tryhackme.com/r/room/networkingsecureprotocols)

Room Prerequisites

This room expects that you know terms such as IP address and TCP port number; however, we don’t expect that the reader is able to explain such terms in proper technical depth. If you are unfamiliar with these terms, please consider joining the [Pre Security](https://tryhackme.com/r/path/outline/presecurity) path.

Learning Objectives

By the time you finish this room, you will have learned about the following:

* ISO OSI network model
* IP addresses, subnets, and routing
* TCP, UDP, and port numbers
* How to connect to an open TCP port from the command line

Task 02:

Before we start, we should note that the OSI model might initially seem complicated. Don’t worry if you encounter cryptic acronyms, as we provide examples of the OSI model layers. We assure you that by the time you finish this module, this task will feel like a piece of cake.

The OSI (Open Systems Interconnection) model is a conceptual model developed by the International Organization for Standardization (ISO) that describes how communications should occur in a computer network. In other words, the OSI model defines a framework for computer network communications. Although this model is theoretical, it is vital to learn and understand as it helps grasp networking concepts on a deeper level. The OSI model is composed of seven layers:

1. Physical Layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

The numbering starts with the physical layer being layer 1, while the top layer, the application layer, is layer 7. To help you remember the layers from bottom to top, you can use a mnemonic such as “Please Do Not Throw Spinach Pizza Away.” You can check the Internet for other easy-to-remember acronyms if this helps you memorise them. Remembering the OSI model layers with their layer numbers is important; otherwise, you will struggle to understand terms such as “layer 3 switch” or “layer 7 firewall.”

Layer 1: Physical Layer

The physical layer, also referred to as layer 1, deals with the physical connection between devices; this includes the medium, such as a wire, and the definition of the binary digits 0 and 1. Data transmission can be via an electrical, optical, or wireless signal. Consequently, we need data cables or antennas, depending on our physical medium.

In addition to Ethernet cable, shown in the illustration below, and optical fibre cable, examples of the physical layer medium include the WiFi radio bands, the 2.4 GHz band, the 5 GHz band, and the 6 GHz band.

Layer 2: Data Link Layer

The physical layer defines a medium to transmit our signal. The data link layer, i.e., layer 2, represents the protocol that enables data transfer between nodes on the same network segment. Let’s put it in simpler terms. The data link layer describes an agreement between the different systems on the same network segment on how to communicate. A network segment refers to a group of networked devices using a shared medium or channel for information transfer. For example, consider a company office with ten computers connected to a network switch; that’s a network segment.

Examples of layer 2 include Ethernet, i.e., 802.3, and WiFi, i.e., 802.11. Ethernet and WiFi addresses are six bytes. Their address is called a MAC address, where MAC stands for Media Access Control. They are usually expressed in hexadecimal format with a colon separating each two bytes. The three leftmost bytes identify the vendor.

We expect to see two MAC addresses in each frame in real network communication over Ethernet or WiFi. The packet in the screenshot below shows:

* The destination data-link address (MAC address) highlighted in yellow
* The source data link address (MAC address) is highlighted in blue
* The remaining bits show the data being sent

Layer 3: Network Layer

The data link layer focuses on sending data between two nodes on the same network segment. The network layer, i.e., layer 3, is concerned with sending data between different networks. In more technical terms, the network layer handles logical addressing and routing, i.e., finding a path to transfer the network packets between the diverse networks.

In the data link layer, we gave an example of one company office with ten computers, where the data link layer is responsible for providing a connection between them. Let’s say that this company has multiple offices distributed across various cities, countries, or even continents. The network layer is responsible for connecting the different offices together.

The network below shows that computers A and B are connected, although on different networks. You can also notice two paths connecting the two computers; the network layer will route the network packets through the path it deems better.

Examples of the network layer include Internet Protocol (IP), Internet Control Message Protocol (ICMP), and Virtual Private Network (VPN) protocols such as IPSec and SSL/TLS VPN.

Layer 4: Transport Layer

Layer 4, the transport layer, enables end-to-end communication between running applications on different hosts. Your web browser is connected to the TryHackMe web server over the transport layer, which can support various functions like flow control, segmentation, and error correction.

Examples of layer 4 are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

Layer 5: Session Layer

The session layer is responsible for establishing, maintaining, and synchronising communication between applications running on different hosts. Establishing a session means initiating communication between applications and negotiating the necessary parameters for the session. Data synchronisation ensures that data is transmitted in the correct order and provides mechanisms for recovery in case of transmission failures.

Examples of the session layer are Network File System (NFS) and Remote Procedure Call (RPC).

Layer 6: Presentation Layer

The presentation layer ensures the data is delivered in a form the application layer can understand. Layer 6 handles data encoding, compression, and encryption. An example of encoding is character encoding, such as ASCII or Unicode.

Various standards are used at the presentation layer. Consider the scenario where we want to send an image via email. First, we use JPEG, GIF, and PNG to save our images; furthermore, although hidden from the user by the email client, we use MIME (Multipurpose Internet Mail Extensions) to attach the file to our email. MIME encodes a binary file using 7-bit ASCII characters.

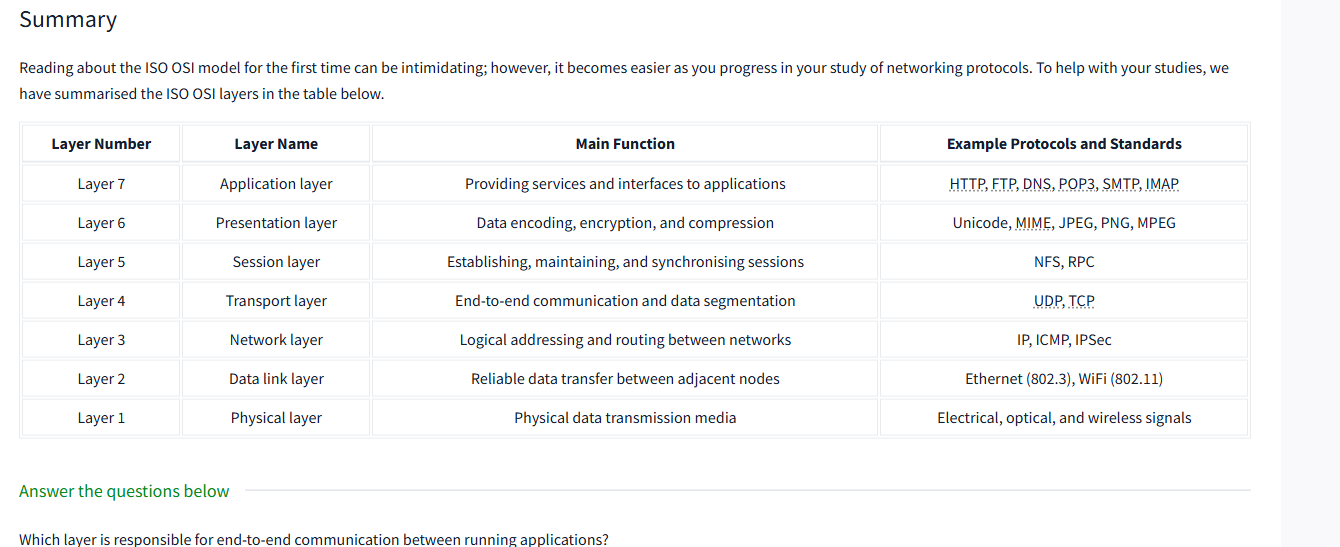
Layer 7: Application Layer

The application layer provides network services directly to end-user applications. Your web browser would use the HTTP protocol to request a file, submit a form, or upload a file.

The application layer is the top layer, and you might have encountered many of its protocols as you use different applications. Examples of Layer 7 protocols are HTTP, FTP, DNS, POP3, SMTP, and IMAP. Don’t worry if you are not familiar with all of them.

Summary

Reading about the ISO OSI model for the first time can be intimidating; however, it becomes easier as you progress in your study of networking protocols. To help with your studies, we have summarised the ISO OSI layers in the table below.



Task 3TCP/IP Model

Now that we have covered the conceptual ISO OSI model, it is time to study an implemented model, the TCP/IP model. TCP/IP stands for Transmission Control Protocol/Internet Protocol and was developed in the 1970s by the Department of Defense (DoD). I hear you ask why DoD would create such a model. One of the strengths of this model is that it allows a network to continue to function as parts of it are out of service, for instance, due to a military attack. This capability is possible in part due to the design of the routing protocols to adapt as the network topology changes.

In our presentation of the ISO OSI model, we went from bottom to top, from layer 1 to layer 7. In this task, let’s look at things from a different perspective, from top to bottom. From top to bottom, we have:

* **Application Layer**: The OSI model application, presentation and session layers, i.e., layers 5, 6, and 7, are grouped into the application layer in the TCP/IP model.
* **Transport Layer**: This is layer 4.
* **Internet Layer**: This is layer 3. The OSI model’s network layer is called the Internet layer in the TCP/IP model.
* **Link Layer**: This is layer 2.

The table below shows how the TCP/IP model layers map to the ISO/OSI model layers.

Here are the correct answers based on the new image:

1. **To which layer does HTTP belong in the TCP/IP model?**  
   ✅ **Application Layer**
2. **How many layers of the OSI model does the application layer in the TCP/IP model cover?**  
   ✅ **Three**

(It covers the **Application**, **Presentation**, and **Session** layers of the OSI model.)

Let me know if you'd like a quick diagram or chart comparing OSI vs. TCP/IP layers!

TASK 04:

Task 4IP Addresses and Subnets

When you hear the word IP address, you might think of an address like 192.168.0.1 or something less common, such as 172.16.159.243. In both cases, you are right. Both of these are IP addresses; IPv4 (IP version 4) addresses to be specific.

Every host on the network needs a unique identifier for other hosts to communicate with him. Without a unique identifier, the host cannot be found without ambiguity. When using the TCP/IP protocol suite, we need to assign an IP address for each device connected to the network.

One analogy of an IP address is your home postal address. Your postal address allows you to receive letters and parcels from all over the world. Furthermore, it can identify your home without ambiguity; otherwise, you cannot shop online!

As you might already know, we have IPv4 and IPv6 (IP version 6). IPv4 is still the most common, and whenever you come across a text mentioning IP without the version, we expect them to mean IPv4.

So, what makes an IP address? An IP address comprises four octets, i.e., 32 bits. Being 8 bits, an octet allows us to represent a decimal number between 0 and 255. An IP address is shown in the image below.



At the risk of oversimplifying things, the 0 and 255 are reserved for the network and broadcast addresses, respectively. In other words, 192.168.1.0 is the network address, while 192.168.1.255 is the broadcast address. Sending to the broadcast address targets all the hosts on the network. With simple math, you can conclude that we cannot have more than 4 billion unique IPv4 addresses. If you are curious about the math, it is approximately 232 because we have 32 bits. This number is approximate because we didn’t consider network and broadcast addresses.

Looking Up Your Network Configuration

You can look up your IP address on the MS Windows command line using the command ipconfig. On Linux and UNIX-based systems, you can issue the command ifconfig or ip address show, which can be typed as ip a s. In the terminal window below, we show ifconfig.

Terminal

**user@TryHackMe$ ifconfig**

**[...]**

**wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500**

**inet 192.168.66.89 netmask 255.255.255.0 broadcast 192.168.66.255**

**inet6 fe80::73e1:ca5e:3f93:b1b3 prefixlen 64 scopeid 0x20<link>**

**ether cc:5e:f8:02:21:a7 txqueuelen 1000 (Ethernet)**

**RX packets 19684680 bytes 18865072842 (17.5 GiB)**

**RX errors 0 dropped 364 overruns 0 frame 0**

**TX packets 14439678 bytes 8773200951 (8.1 GiB)**

**TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0**

The terminal output above indicates the following:

* The host (laptop) IP address is 192.168.66.89
* The subnet mask is 255.255.255.0
* The broadcast address is 192.168.66.255

Let’s use ip a s to compare how the network card IP address is presented.

Terminal

**user@TryHackMe$ ip a s**

**[...]**

**4: wlo1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000**

**link/ether cc:5e:f8:02:21:a7 brd ff:ff:ff:ff:ff:ff**

**altname wlp3s0**

**inet 192.168.66.89/24 brd 192.168.66.255 scope global dynamic noprefixroute wlo1**

**valid\_lft 36795sec preferred\_lft 36795sec**

**inet6 fe80::73e1:ca5e:3f93:b1b3/64 scope link noprefixroute**

**valid\_lft forever preferred\_lft forever**

The terminal output above indicates the following:

* The host (laptop) IP address is 192.168.66.89/24
* The broadcast address is 192.168.66.255

If you are wondering, a subnet mask of 255.255.255.0 can also be written as /24. The /24 means that the leftmost 24 bits within the IP address do not change across the network, i.e., the subnet. In other words, the leftmost three octets are the same across the whole subnet; therefore, we can expect to find addresses that range from 192.168.66.1 to 192.168.66.254. Similar to what was mentioned earlier, 192.168.66.0 and 192.168.66.255 are the network and broadcast addresses, respectively.

Private Addresses

As we are explaining IP addresses, it is useful to mention that for most practical purposes, there are two types of IP addresses:

* Public IP addresses
* Private IP addresses

RFC 1918 defines the following three ranges of private IP addresses:

* 10.0.0.0 - 10.255.255.255 (10/8)
* 172.16.0.0 - 172.31.255.255 (172.16/12)
* 192.168.0.0 - 192.168.255.255 (192.168/16)

We presented earlier an analogy stating that a public IP address is like your home postal address. A private IP address is different; the original idea is that it cannot reach or be reached from the outside world. It is like an isolated city or a compound, where all houses and apartments are numbered systematically and can easily exchange mail with each other, but not with the outside world. For a private IP address to access the Internet, the router must have a public IP address and must support Network Address Translation (NAT). At this stage, let’s not worry about understanding how NAT works, as we will revisit it later in this module.

Before moving on, I recommend memorising the private IP address ranges. Otherwise, you might see an IP address such as 10.1.33.7 or 172.31.33.7 and try to access it from a public IP address.

Routing

A router is like your local post office; you hand them the mail parcel, and they would know how to deliver it. If we dig deeper, you might mail something to an address in another city or country. The post office will check the address and decide where to send it next. For example, if it is to leave the country, we expect one central office to handle all shipments abroad.

In technical terms, a router forwards data packets to the proper network. Usually, a data packet passes through multiple routers before it reaches its final destination. The router functions at layer 3, inspecting the IP address and forwarding the packet to the best network (router) so the packet gets closer to its destination.

Answer the questions below

Which of the following IP addresses is not a private IP address?

* 192.168.250.125
* 10.20.141.132
* 49.69.147.197
* 172.23.182.251

**❓ Which of the following IP addresses is *not* a private IP address?**

**Private IP address ranges are:**

| **Class** | **Range** |
| --- | --- |
| A | 10.0.0.0 – 10.255.255.255 |
| B | 172.16.0.0 – 172.31.255.255 |
| C | 192.168.0.0 – 192.168.255.255 |

Let's evaluate each:

1. **192.168.250.125** ✅  
   → Falls in the **192.168.x.x** private range (Class C) → **Private**
2. **10.20.141.132** ✅  
   → Falls in the **10.x.x.x** private range (Class A) → **Private**
3. **49.69.147.197** ❌  
   → This is **not in any private range**, it's a **public IP** → ✅ **Correct Answer**
4. **172.23.182.251** ✅  
   → Falls in the **172.16.x.x to 172.31.x.x** range → **Private**

**✅ Answer:** 49.69.147.197  
**💡 Reason:** It doesn’t fall within any of the three private IP ranges.

**❓ Which of the following IP addresses is *not* a valid IP address?**

Each **IPv4 address** consists of 4 octets (numbers), **each ranging from 0 to 255**.

Let's evaluate each:

1. **192.168.250.15** ✅  
   → All octets are within range (0–255) → **Valid**
2. **192.168.254.17** ✅  
   → All octets are within range → **Valid**
3. **192.168.305.19** ❌  
   → **305** is greater than **255**, which is **invalid** → ✅ **Correct Answer**
4. **192.168.199.13** ✅  
   → All values are within the valid range → **Valid**

**✅ Answer:** 192.168.305.19  
**💡 Reason:** The second octet 305 exceeds the valid limit (0–255) for IPv4 addresses.

Task 05:

UDP and TCP

The IP protocol allows us to reach a destination host on the network; the host is identified by its IP address. We need protocols that would enable processes on networked hosts to communicate with each other. There are two transport protocols to achieve that: UDP and TCP.

UDP

UDP (User Datagram Protocol) allows us to reach a specific process on this target host. UDP is a simple connectionless protocol that operates at the transport layer, i.e., layer 4. Being connectionless means that it does not need to establish a connection. UDP does not even provide a mechanism to know that the packet has been delivered.

An IP address identifies the host; we need a mechanism to determine the sending and receiving process. This can be achieved by using port numbers. A port number uses two octets; consequently, it ranges between 1 and 65535; port 0 is reserved. (The number 65535 is calculated by the expression 216 − 1.)

A real-life example similar to UDP is the standard mail service, with no delivery confirmation. In other words, there is no guarantee that the UDP packet has been received successfully, similar to the case of sending a parcel using standard mail with no confirmation of delivery. In the case of standard mail, it means a cheaper cost than the mail delivery options with confirmation. In the case of UDP, it means better speed than a transport protocol that provides “confirmation.”

But what if we want a transport protocol that acknowledges received packets? The answer lies in using TCP instead of UDP.

TCP

TCP (Transmission Control Protocol) is a connection-oriented transport protocol. It uses various mechanisms to ensure reliable data delivery sent by the different processes on the networked hosts. Like UDP, it is a layer 4 protocol. Being connection-oriented, it requires the establishment of a TCP connection before any data can be sent.

In TCP, each data octet has a sequence number; this makes it easy for the receiver to identify lost or duplicated packets. The receiver, on the other hand, acknowledges the reception of data with an acknowledgement number specifying the last received octet.

A TCP connection is established using what’s called a three-way handshake. Two flags are used: SYN (Synchronise) and ACK (Acknowledgment). The packets are sent as follows:

1. SYN Packet: The client initiates the connection by sending a SYN packet to the server. This packet contains the client’s randomly chosen initial sequence number.
2. SYN-ACK Packet: The server responds to the SYN packet with a SYN-ACK packet, which adds the initial sequence number randomly chosen by the server.
3. ACK Packet: The three-way handshake is completed as the client sends an ACK packet to acknowledge the reception of the SYN-ACK packet.

Similar to UDP, TCP identifies the process of initiating or waiting (listening) for a connection using port numbers. As stated, a valid port number ranges between 1 and 65535 because it uses two octets and port 0 is reserved.

**Question 1: Which protocol requires a three-way handshake?**

**Answer:** TCP (Transmission Control Protocol)

**Reasoning:**

* **Connection-Oriented:** TCP is a connection-oriented protocol, meaning it establishes a reliable connection before data transfer begins. This connection setup is crucial for ensuring data integrity and order.
* **Three-Way Handshake Purpose:** The three-way handshake is the method TCP uses to initiate this reliable connection. It ensures both the client and server are ready to communicate and agree on initial sequence numbers.
* **Steps of the Handshake:**
  1. **SYN (Synchronize):** The client sends a SYN packet to the server, indicating its desire to establish a connection.
  2. **SYN-ACK (Synchronize-Acknowledge):** The server responds with a SYN-ACK packet, acknowledging the client's 1 SYN and sending its own SYN.

[1. medium.com](https://medium.com/@lihua.long/%E5%B5%8C%E5%85%A5%E5%BC%8F%E8%BD%AF%E4%BB%B6%E9%9D%A2%E8%AF%95%E7%BB%8F%E5%85%B8%E9%97%AE%E9%A2%98-89f48e0d5aaf" \t "_blank)

[medium.com](https://medium.com/@lihua.long/%E5%B5%8C%E5%85%A5%E5%BC%8F%E8%BD%AF%E4%BB%B6%E9%9D%A2%E8%AF%95%E7%BB%8F%E5%85%B8%E9%97%AE%E9%A2%98-89f48e0d5aaf" \t "_blank)

* 1. **ACK (Acknowledge):** The client sends an ACK packet, acknowledging the server's SYN-ACK.

Only after these three steps are completed is the connection established, and data transfer can begin.

**Question 2: What is the approximate number of port numbers (in thousands)?**

**Answer:** 65

**Reasoning:**

* **Port Number Range:** Port numbers are 16-bit unsigned integers, meaning they can range from 0 to 65535.
* **Approximation in Thousands:** To express this in thousands, we divide 65535 by 1000, which gives us approximately 65.535.
* **Rounded Value:** Since the question asks for the "approximate" number in thousands, we round 65.535 to 65.

Therefore, there are approximately 65,000 port numbers (or 65 thousand).

**❓ Which of the following IP addresses is *not* a valid IP address?**

Each **IPv4 address** consists of 4 octets (numbers), **each ranging from 0 to 255**.

Let's evaluate each:

1. **192.168.250.15** ✅  
   → All octets are within range (0–255) → **Valid**
2. **192.168.254.17** ✅  
   → All octets are within range → **Valid**
3. **192.168.305.19** ❌  
   → **305** is greater than **255**, which is **invalid** → ✅ **Correct Answer**
4. **192.168.199.13** ✅  
   → All values are within the valid range → **Valid**

**✅ Answer:** 192.168.305.19  
**💡 Reason:** The second octet 305 exceeds the valid limit (0–255) for IPv4 addresses.