Task 06 :

Task 6Encapsulation

Before wrapping up, it is crucial to explain another key concept: **encapsulation**. In this context, encapsulation refers to the process of every layer adding a header (and sometimes a trailer) to the received unit of data and sending the “encapsulated” unit to the layer below.

Encapsulation is an essential concept as it allows each layer to focus on its intended function. In the image below, we have the following four steps:

* **Application data**: It all starts when the user inputs the data they want to send into the application. For example, you write an email or an instant message and hit the send button. The application formats this data and starts sending it according to the application protocol used, using the layer below it, the transport layer.
* **Transport protocol segment or datagram**: The transport layer, such as TCP or UDP, adds the proper header information and creates the TCP **segment** (or UDP **datagram**). This segment is sent to the layer below it, the network layer.
* **Network packet**: The network layer, i.e. the Internet layer, adds an IP header to the received TCP segment or UDP datagram. Then, this IP **packet** is sent to the layer below it, the data link layer.
* **Data link frame**: The Ethernet or WiFi receives the IP packet and adds the proper header and trailer, creating a **frame**.

We start with application data. At the transport layer, we add a TCP or UDP header to create a **TCP segment** or **UDP datagram**. Again, at the network layer, we add the proper IP header to get an **IP packet** that can be routed over the Internet. Finally, we add the appropriate header and trailer to get a WiFi or Ethernet frame at the link layer.

The process has to be reversed on the receiving end until the application data is extracted.

The Life of a Packet

Based on what we have studied so far, we can explain a *simplified version* of the packet’s life. Let’s consider the scenario where you search for a room on TryHackMe.

1. On the TryHackMe search page, you enter your search query and hit enter.
2. Your web browser, using HTTPS, prepares an HTTP request and pushes it to the layer below it, the transport layer.
3. The TCP layer needs to establish a connection via a three-way handshake between your browser and the TryHackMe web server. After establishing the TCP connection, it can send the HTTP request containing the search query. Each TCP segment created is sent to the layer below it, the Internet layer.
4. The IP layer adds the source IP address, i.e., your computer, and the destination IP address, i.e., the IP address of the TryHackMe web server. For this packet to reach the router, your laptop delivers it to the layer below it, the link layer.
5. Depending on the protocol, The link layer adds the proper link layer header and trailer, and the packet is sent to the router.
6. The router removes the link layer header and trailer, inspects the IP destination, among other fields, and routes the packet to the proper link. Each router repeats this process until it reaches the router of the target server.

The steps will then be reversed as the packet reaches the router of the destination network. As we cover additional protocols, we will revisit this exercise and create a more in-depth version.

Answer the questions below

On a WiFi, within what will an IP packet be encapsulated?

Top of Form

Bottom of Form

What do you call the UDP data unit that encapsulates the application data?

Top of Form

Bottom of Form

What do you call the data unit that encapsulates the application data sent over TCP?

Let's get straight to the point with these answers:

**Question 1: On a WIFI, within what will an IP packet be encapsulated?**

**Answer:** Ethernet frame (or 802.11 frame)

**Question 2: What do you call the UDP data unit that encapsulates the application data?**

**Answer:** Datagram

**Question 3: What do you call the data unit that encapsulates the application data sent over TCP?Answer:** Segment

Task 6 :

**Encapsulation**

Before wrapping up, it is crucial to explain another key concept: **encapsulation**. In this context, encapsulation refers to the process of every layer adding a header (and sometimes a trailer) to the received unit of data and sending the “encapsulated” unit to the layer below.

Encapsulation is an essential concept as it allows each layer to focus on its intended function. In the image below, we have the following four steps:

* **Application data**: It all starts when the user inputs the data they want to send into the application. For example, you write an email or an instant message and hit the send button. The application formats this data and starts sending it according to the application protocol used, using the layer below it, the transport layer.
* **Transport protocol segment or datagram**: The transport layer, such as TCP or UDP, adds the proper header information and creates the TCP **segment** (or UDP **datagram**). This segment is sent to the layer below it, the network layer.
* **Network packet**: The network layer, i.e. the Internet layer, adds an IP header to the received TCP segment or UDP datagram. Then, this IP **packet** is sent to the layer below it, the data link layer.
* **Data link frame**: The Ethernet or WiFi receives the IP packet and adds the proper header and trailer, creating a **frame**.

We start with application data. At the transport layer, we add a TCP or UDP header to create a **TCP segment** or **UDP datagram**. Again, at the network layer, we add the proper IP header to get an **IP packet** that can be routed over the Internet. Finally, we add the appropriate header and trailer to get a WiFi or Ethernet frame at the link layer.

The process has to be reversed on the receiving end until the application data is extracted.

The Life of a Packet

Based on what we have studied so far, we can explain a *simplified version* of the packet’s life. Let’s consider the scenario where you search for a room on TryHackMe.

1. On the TryHackMe search page, you enter your search query and hit enter.
2. Your web browser, using HTTPS, prepares an HTTP request and pushes it to the layer below it, the transport layer.
3. The TCP layer needs to establish a connection via a three-way handshake between your browser and the TryHackMe web server. After establishing the TCP connection, it can send the HTTP request containing the search query. Each TCP segment created is sent to the layer below it, the Internet layer.
4. The IP layer adds the source IP address, i.e., your computer, and the destination IP address, i.e., the IP address of the TryHackMe web server. For this packet to reach the router, your laptop delivers it to the layer below it, the link layer.
5. Depending on the protocol, The link layer adds the proper link layer header and trailer, and the packet is sent to the router.
6. The router removes the link layer header and trailer, inspects the IP destination, among other fields, and routes the packet to the proper link. Each router repeats this process until it reaches the router of the target server.

The steps will then be reversed as the packet reaches the router of the destination network. As we cover additional protocols, we will revisit this exercise and create a more in-depth version.

Top of Form