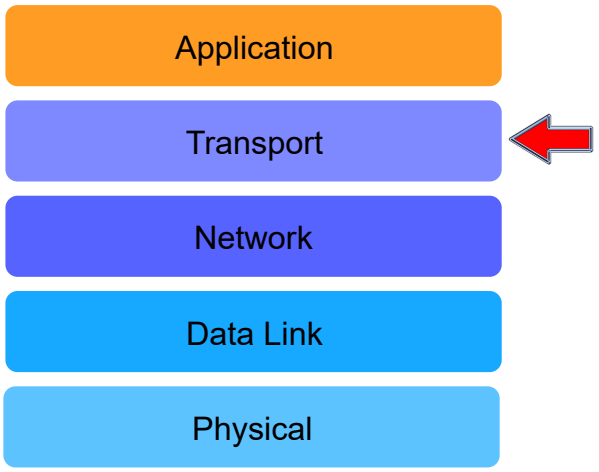
What Is the Transport Layer?

We finished the application layer, and now we'll study the transport layer.

WE'LL COVER THE FOLLOWING

You Are Here!

Let's zoom out and have a look at the big picture.



Key Responsibilities of the Transport Layer

- **Extends network to the applications**: the transport layer takes messages from the network to applications. In other words, while the network layer (directly below the transport layer) transports messages from one end-system to another, the transport layer delivers the message to and from the relevant application *on* an end-system.

The Post Analogy

Imagine you post a package across the world. Presumably, a ship or an airplane would carry the message to the relevant country. However, the post system of that country would take it to the relevant address. The **plane/ship** is the *network layer* and the **post system** is the *transport layer*.

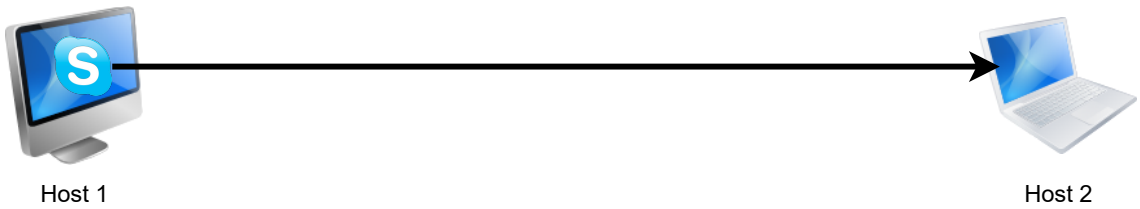
Have a look at the slides for a clearer explanation.

suppose two end-systems are communicating with each other on Skype via the internet

1 of 4

and one end-system sends off a message to the other

2 of 4



the network layer will handle delivering the message to the host

3 of 4

however, delivering the message to the relevant application on the other end system is handled by the transport layer

4 of 4

Here are some other responsibilities of the transport layer.

- **Logical application-to-application delivery**, the transport layer makes it so that applications can address other applications on other end-systems directly. This is true even if it exists halfway across the world. So it provides a layer of **abstraction**.
- **Segments data**. The transport layer also divides the data into manageable pieces called ‘segments’ or ‘datagrams.’
- **Can allow multiple conversations**. Tracks each application to application connection or ‘conversation’ separately, which can allow multiple conversations to occur at once.
- **Multiplexes & demultiplexes data**. It ensures that the data reaches the relevant application *within* an end-system. So if multiple packets get sent to one host, each will end up at the correct application.

Where It Exists

- The transport layer does not have anything to do with the **core of the network**. Its only responsibility is to take messages from an *application* on a machine and hand them off to the network layer. The network layer transfers messages from one host to another.

- The transport layer also receives messages from the network layer and transports them to the correct application.

Therefore, the transport layer and its protocols **reside on end-systems**! It is also the first layer in the OSI reference model (from the bottom) that distinguishes between applications.

Transport Layer Protocols

The transport layer has two prominent protocols: the **transmission control protocol** and the **user datagram protocol**. In general, an application developer will have to choose between the two. We'll discuss the intricacies of each in detail in upcoming chapters, but here is a quick overview.

TCP

- Delivers messages that we call 'segments' reliably and in order.
- Detects any modifications that may have been introduced in the packets during delivery and corrects them.
- Handles the volumes of traffic at one time within the network core by sending only an appropriate amount of data at one time.
- Examples of applications/application protocols that use TCP are: HTTP, E-mail, File Transfers.

UDP

- Does not ensure in-order delivery of messages that we call 'datagrams.'
- Detects any modifications that may have been introduced in the packets during delivery but does not correct them by default.
- Does not ensure reliable delivery.
- Generally faster than TCP because of the reduced overhead of ensuring uncorrupted delivery of packets in order.
- Applications that use UDP include: Domain Name System (DNS), live video streaming, and Voice over IP (VoIP).

Quick Quiz!

1 The transport layer in the OSI reference model uses the services of _____ layer.

COMPLETED 0%

1 of 2

In the next lesson, we'll have a more in-depth look at multiplexing and demultiplexing!