The Transmission Control Protocol

In this lesson, we'll look at a quick overview of TCP and some famous applications that use it.

Introduction to TCP

TCP, or the **transmission control protocol**, is one of the two key protocols of the transport layer. TCP is what makes most modern applications as enjoyable and reliable as they are. HTTP's implementation, for example, would be very complex, if it weren't for TCP.

TCP is a robust protocol meant to adapt to a diverse range of network topologies, bandwidths, delays, message sizes, and other varying factors that exist in the network layer.

An Analogy: Talking on a Cell Phone

TCP is a **connection-oriented protocol** unlike UDP. The connection orientedness is like a **phone call** because a connection is established before communication takes place, and then we hang up. Some scenarios that might occur on a phone call with a friend are relevant to connection-oriented
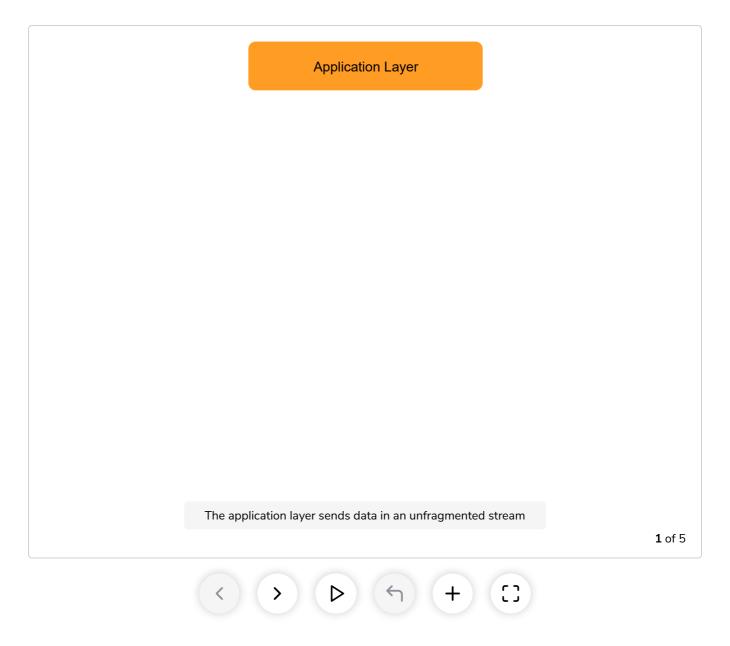
protocols in the Internet, too. For example,

- What if you **can't understand** what your friend is saying?
  - Should you ask them to **repeat what they said**?
- What if you haven't **heard them speak for a while**?
  - Does your friend simply **not have anything to say**?
  - Has the phone lost reception and **disconnected**?
  - Should you **keep talking**? If so, **for how long**?
  - Should both you and your friend periodically say "mhmm" to indicate that you are present and listening?

TCP, being THE connection-oriented transport layer protocol for the Internet, has mechanisms to solves these problems.

What TCP Does

Here are some key responsibilities of the protocol.

1. **Send data** at an appropriate transmission rate. It should be a fast enough rate to make full use of the available capacity but it shouldn't be so fast as to cause congestion.

2. **Segment data.** The application layer sends the transport layer a continuous and unsegmented stream of data so that there's no limit to how much data the application layer can give to the transport layer at once. Hence, the transport layer divides it into appropriately sized **segments**. Note that a segment is a collection of bytes. Furthermore, when a TCP segment is too big, the network layer may break it into multiple network layer messages, so the receiving TCP entity would have to re-assemble the network layer messages.

3. **End to end flow control**. Flow control means **not overwhelming the receiver**. It's not the same as congestion control. Congestion control tries not to choke the network. However, if the receiving machine is slow, it might drown in data even if the network is not choked. Avoiding drowning the receiver in data is end to end flow control. There is also hop by hop flow control, which is done at the data link layer.

3. **Identify and retransmit messages** that do not get delivered. The network layer cannot be relied upon to deliver messages.

4. **Identify when messages are received out of order and reassemble** them. The network layer can also *not* be relied upon to transmit messages in order.

Well-Known Applications That Use TCP

File Transfer

FTP or **File Transfer Protocol** is built on top of TCP. It uses ports **20** and **21**. When transferring files, we wouldn't want some bytes of the file completely missing, or some chunks in the file re-ordered or some byte values changed during transfer. That's why TCP is a natural choice for FTP. In other words, it uses TCP for its **reliability**, which is a key part of file transfer.

Secure Shell `SSH`

SSH or **Secure Shell** is a protocol to allow a secure connection to a remote host over an unsecured network. It's widely popular and most programmers use it to date to execute operating system shell commands on remote servers. The reasons that this application uses TCP is similar to FTP, and that's reliable delivery.
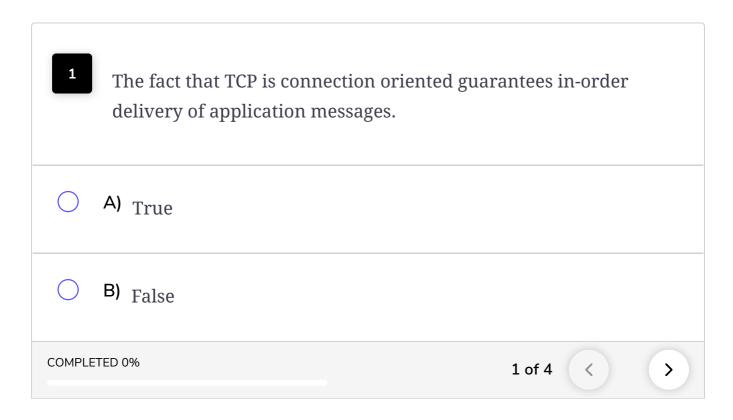
Email

All email protocols, SMTP, IMAP, and POP use TCP to ensure complete and reliable message delivery similar to the reasons that FTP uses TCP.

Web Browsing

Web browsing on both HTTP and HTTPS is done on TCP as well for the same reasons as FTP.

Quick Quiz!

---

1   The fact that TCP is connection oriented guarantees in-order delivery of application messages.

○ A) True

○ B) False

COMPLETED 0%   1 of 4   ‹   ›

Now that we have an overview of what TCP is and what it does, let's look at some of its key properties in the next lesson!