The OSI Model

Learning Objectives

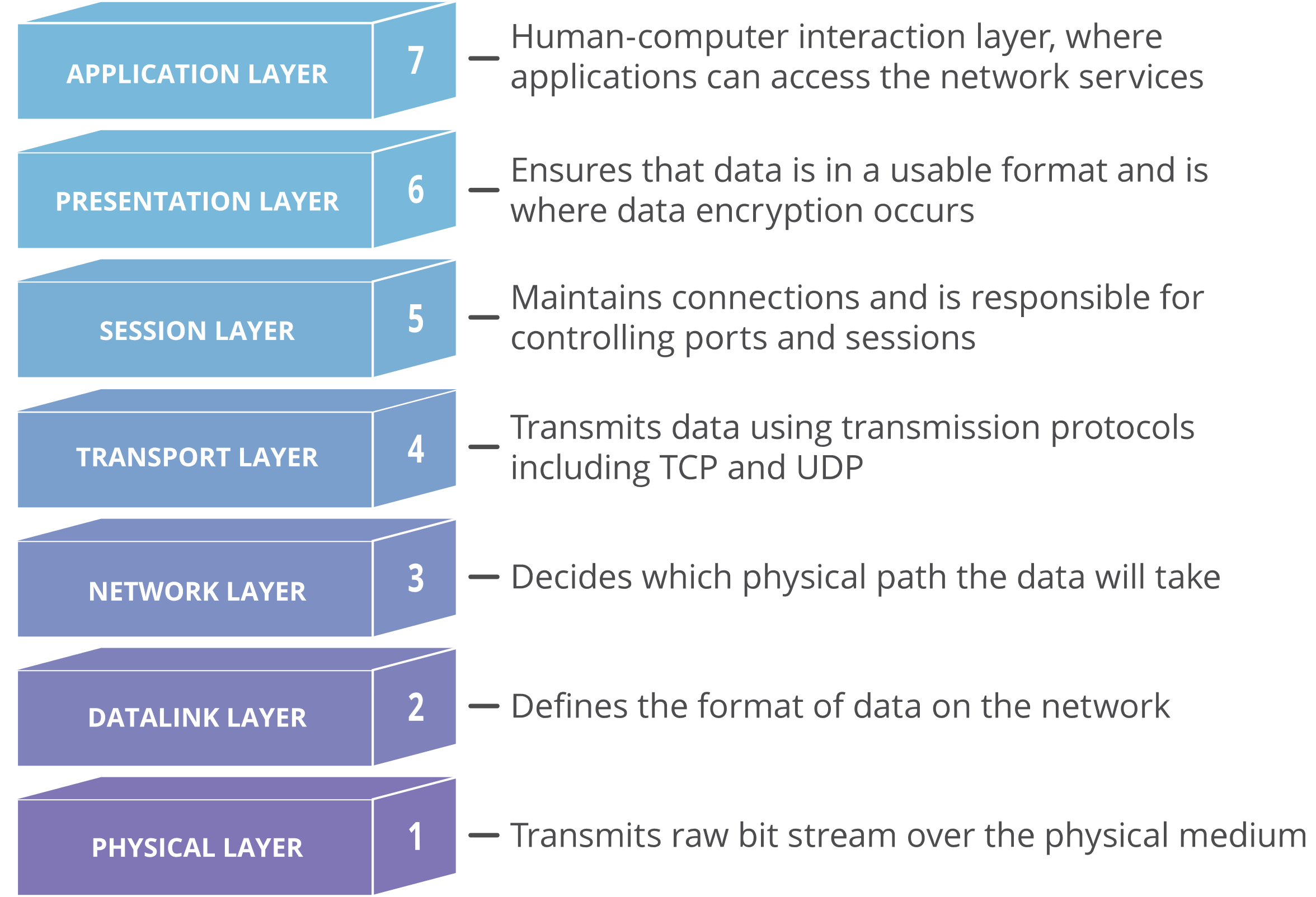
After reading this article you will be able to:

* Define the OSI model
* Identify the 7 layers of the OSI model
* Explore how data flows through the OSI model

## What is the OSI model?

The Open Systems Interconnection (OSI) model is a conceptual model created by the International Organization for Standardization which enables diverse communication systems to communicate using standard protocols. In plain English, the OSI provides a standard for different computer systems to be able to communicate with each other.

The OSI model can be seen as a universal language for computer networking. It’s based on the concept of splitting up a communication system into seven abstract layers, each one stacked upon the last.



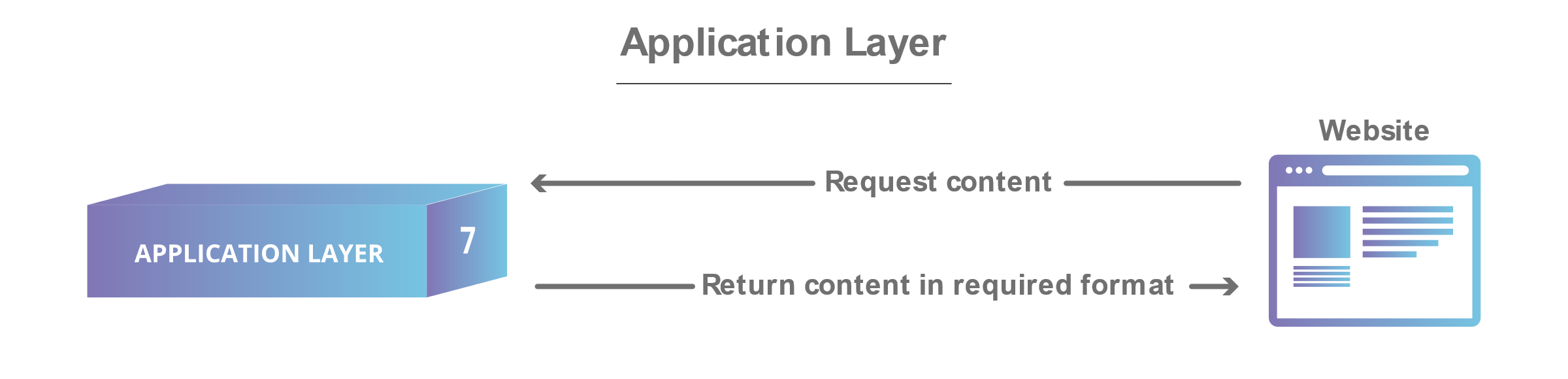
Each layer of the OSI model handles a specific job and communicates with the layers above and below itself. [DDoS attacks](https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/) target specific layers of a network connection; [application layer attacks](https://www.cloudflare.com/learning/ddos/application-layer-ddos-attack/) target [layer 7](https://www.cloudflare.com/learning/ddos/what-is-layer-7/) and protocol layer attacks target layers 3 and 4.

## Why does the OSI model matter?

Although the modern Internet doesn’t strictly follow the OSI model (it more closely follows the simpler Internet protocol suite), the OSI model is still very useful for troubleshooting network problems. Whether it’s one person who can’t get their laptop on the Internet, or a web site being down for thousands of users, the OSI model can help to break down the problem and isolate the source of the trouble. If the problem can be narrowed down to one specific layer of the model, a lot of unnecessary work can be avoided.

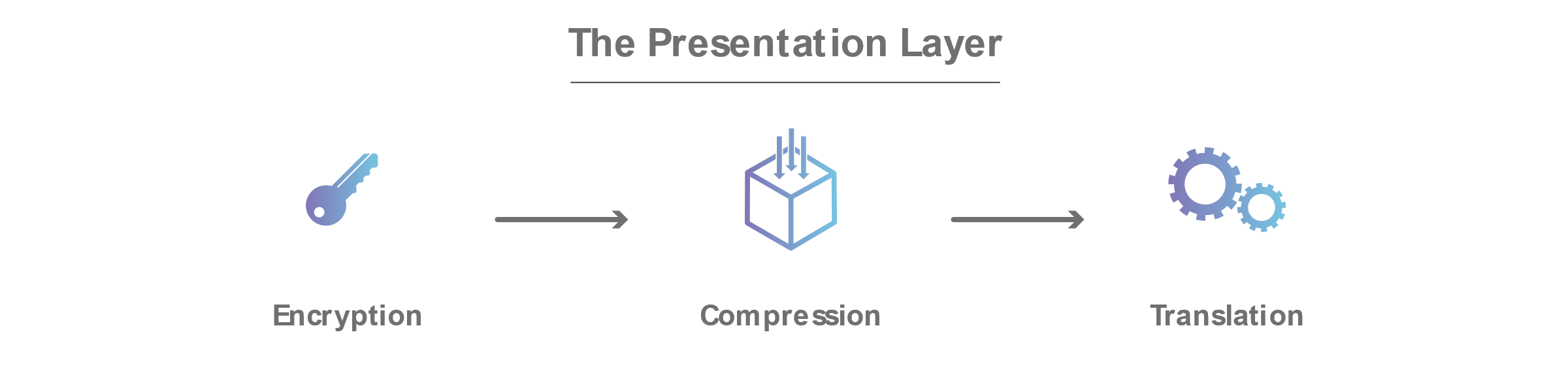
## What are the seven layers of the OSI model?

The seven abstraction layers of the OSI model can be defined as follows, from top to bottom:



#### 7. The Application Layer

This is the only layer that directly interacts with data from the user. Software applications like web browsers and email clients rely on the application layer to initiate communications. But it should be made clear that client software applications are not part of the application layer; rather the application layer is responsible for the protocols and data manipulation that the software relies on to present meaningful data to the user. Application layer protocols include [HTTP](https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/) as well as SMTP (Simple Mail Transfer Protocol is one of the protocols that enables email communications).



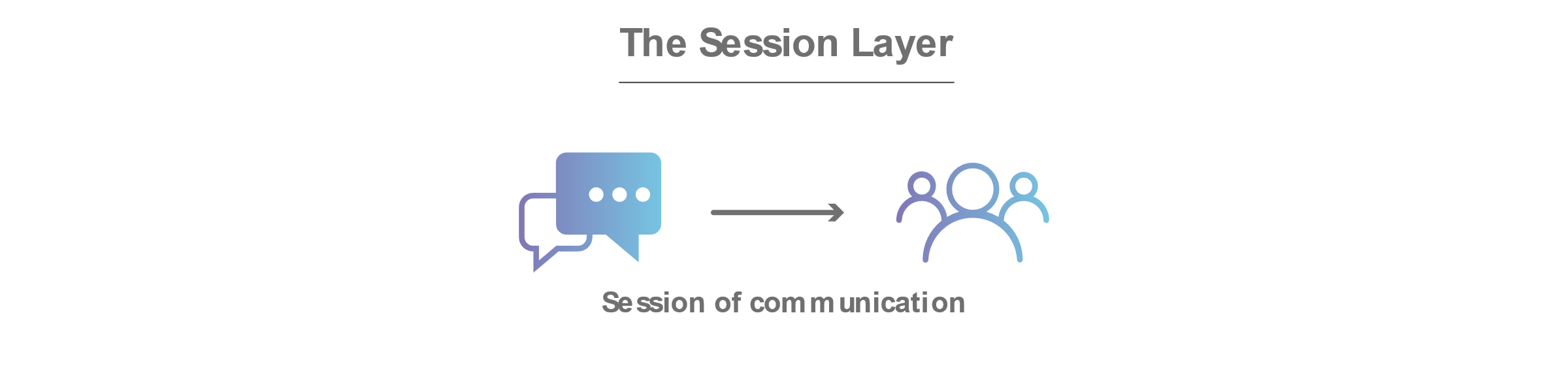
#### 6. The Presentation Layer

This layer is primarily responsible for preparing data so that it can be used by the application layer; in other words, layer 6 makes the data presentable for applications to consume. The presentation layer is responsible for translation, [encryption](https://www.cloudflare.com/learning/ssl/what-is-encryption/), and compression of data.

Two communicating devices communicating may be using different encoding methods, so layer 6 is responsible for translating incoming data into a syntax that the application layer of the receiving device can understand.

If the devices are communicating over an encrypted connection, layer 6 is responsible for adding the encryption on the sender’s end as well as decoding the encryption on the receiver's end so that it can present the application layer with unencrypted, readable data.

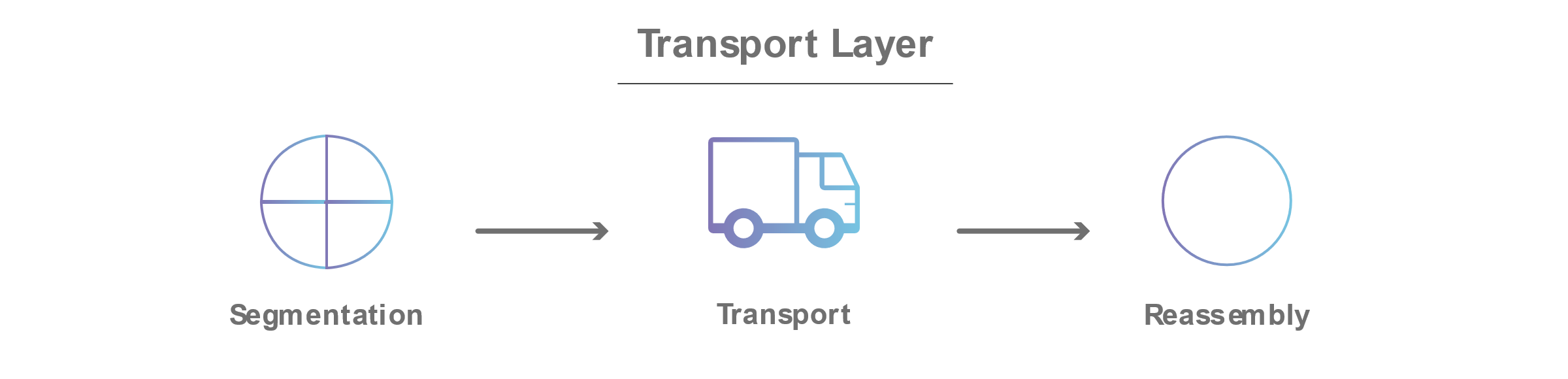
Finally the presentation layer is also responsible for compressing data it receives from the application layer before delivering it to layer 5. This helps improve the speed and efficiency of communication by minimizing the amount of data that will be transferred.



#### 5. The Session Layer

This is the layer responsible for opening and closing communication between the two devices. The time between when the communication is opened and closed is known as the session. The session layer ensures that the session stays open long enough to transfer all the data being exchanged, and then promptly closes the session in order to avoid wasting resources.

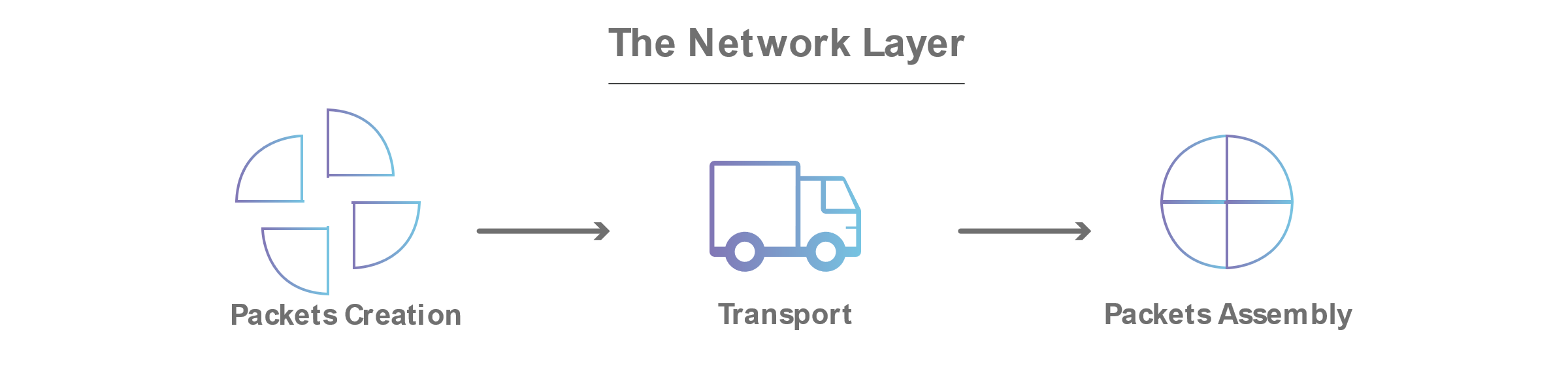
The session layer also synchronizes data transfer with checkpoints. For example, if a 100 megabyte file is being transferred, the session layer could set a checkpoint every 5 megabytes. In the case of a disconnect or a crash after 52 megabytes have been transferred, the session could be resumed from the last checkpoint, meaning only 50 more megabytes of data need to be transferred. Without the checkpoints, the entire transfer would have to begin again from scratch.



#### 4. The Transport Layer

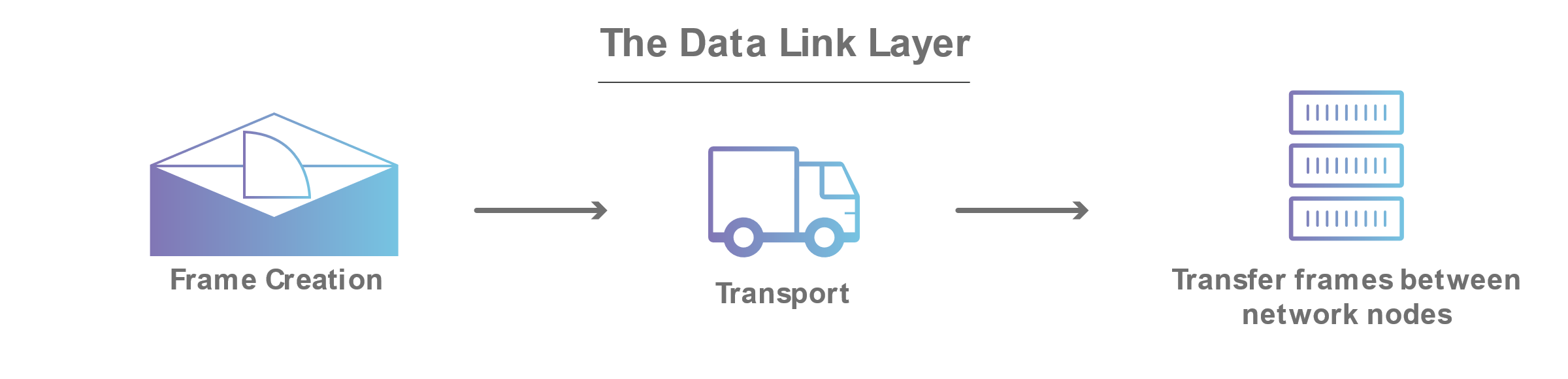
Layer 4 is responsible for end-to-end communication between the two devices. This includes taking data from the session layer and breaking it up into chunks called segments before sending it to layer 3. The transport layer on the receiving device is responsible for reassembling the segments into data the session layer can consume.

The transport layer is also responsible for flow control and error control. Flow control determines an optimal speed of transmission to ensure that a sender with a fast connection doesn’t overwhelm a receiver with a slow connection. The transport layer performs error control on the receiving end by ensuring that the data received is complete, and requesting a retransmission if it isn’t.



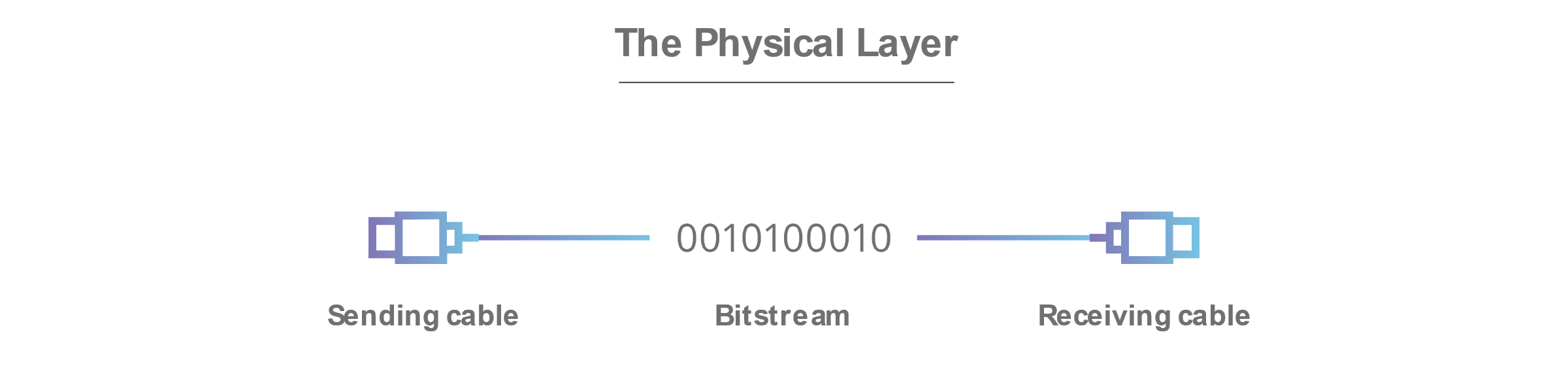
#### 3. The Network Layer

The network layer is responsible for facilitating data transfer between two different networks. If the two devices communicating are on the same network, then the network layer is unnecessary. The network layer breaks up segments from the transport layer into smaller units, called packets, on the sender’s device, and reassembling these packets on the receiving device. The network layer also finds the best physical path for the data to reach its destination; this is known as routing.



#### 2. The Data Link Layer

The data link layer is very similar to the network layer, except the data link layer facilitates data transfer between two devices on the SAME network. The data link layer takes packets from the network layer and breaks them into smaller pieces called frames. Like the network layer, the data link layer is also responsible for flow control and error control in intra-network communication (The transport layer only does flow control and error control for inter-network communications).



#### 1. The Physical Layer

This layer includes the physical equipment involved in the data transfer, such as the cables and switches. This is also the layer where the data gets converted into a bit stream, which is a string of 1s and 0s. The physical layer of both devices must also agree on a signal convention so that the 1s can be distinguished from the 0s on both devices.

## How data flows through the OSI model

In order for human-readable information to be transferred over a network from one device to another, the data must travel down the seven layers of the OSI model on the sending device and then travel up the seven layers on the receiving end.

For example: Mr. Cooper wants to send Ms. Palmer an email. Mr. Cooper composes his message in an email application on his laptop and then hits ‘send’. His email application will pass his email message over to the application layer, which will pick a protocol (SMTP) and pass the data along to the presentation layer. The presentation layer will then compress the data and then it will hit the session layer, which will initialize the communication session.

The data will then hit the sender’s transportation layer where it will be segmented, then those segments will be broken up into packets at the network layer, which will be broken down even further into frames at the data link layer. The data link layer will then deliver those frames to the physical layer, which will convert the data into a bitstream of 1s and 0s and send it through a physical medium, such as a cable.

Once Ms. Palmer’s computer receives the bit stream through a physical medium (such as her wifi), the data will flow through the same series of layers on her device, but in the opposite order. First the physical layer will convert the bitstream from 1s and 0s into frames that get passed to the data link layer. The data link layer will then reassemble the frames into packets for the network layer. The network layer will then make segments out of the packets for the transport layer, which will reassemble the segments into one piece of data.

The data will then flow into the receiver's session layer, which will pass the data along to the presentation layer and then end the communication session. The presentation layer will then remove the compression and pass the raw data up to the application layer. The application layer will then feed the human-readable data along to Ms. Palmer’s email software, which will allow her to read Mr. Cooper’s email on her laptop screen.

## TCP/IP

## Learning Objectives

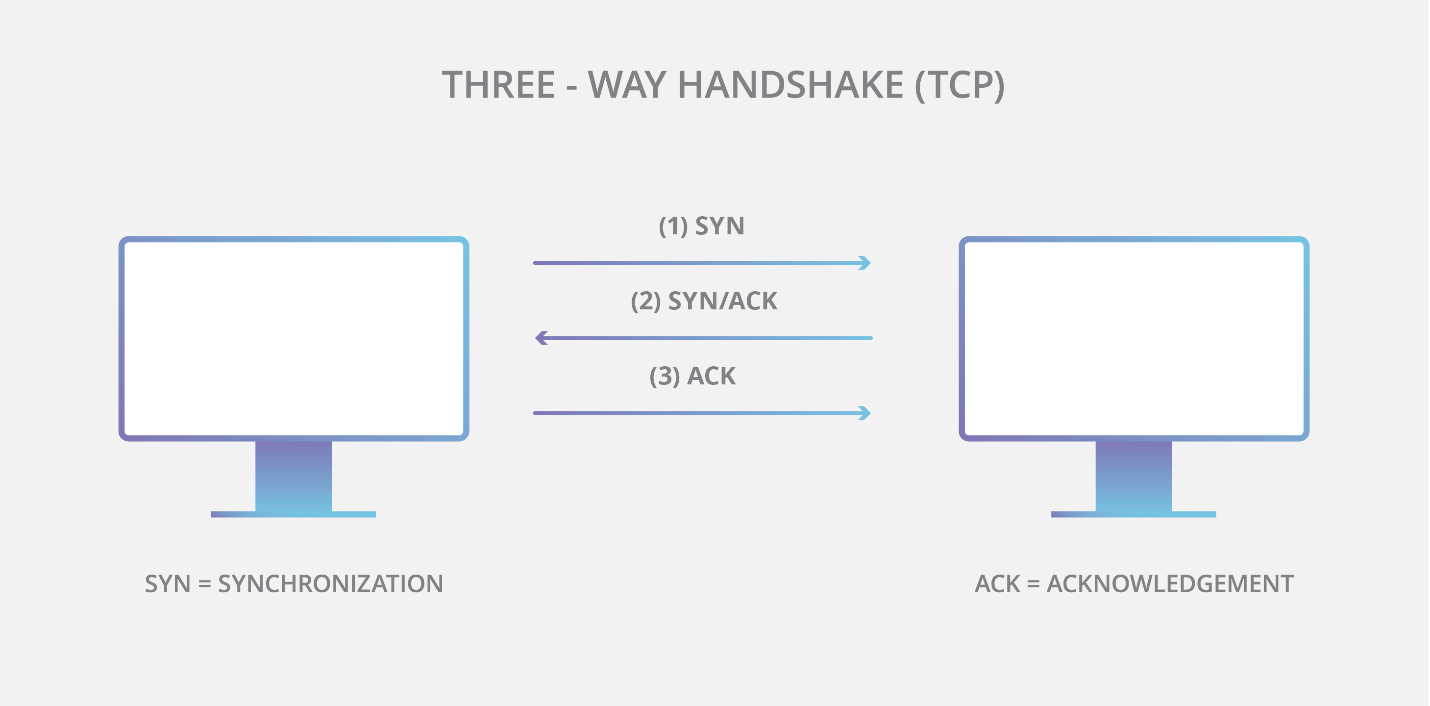
## What are IP & TCP?

The Internet Protocol (IP) is the address system of the Internet and has the core function of delivering packets of information from a source device to a target device. IP is the primary way in which network connections are made, and it establishes the basis of the Internet. IP does not handle packet ordering or error checking. Such functionality requires another protocol, typically TCP.

The TCP/IP relationship is similar to sending someone a message written on a puzzle through the mail. The message is written down and the puzzle is broken into pieces. Each piece then can travel through a different postal route, some of which take longer than others. When the puzzle pieces arrive after traversing their different paths, the pieces may be out of order. The Internet Protocol makes sure the pieces arrive at their destination address. The TCP protocol can be thought of as the puzzle assembler on the other side who puts the pieces together in the right order, asks for missing pieces to be resent, and lets the sender know the puzzle has been received. TCP maintains the connection with the sender from before the first puzzle piece is sent to after the final piece is sent.

IP is a connectionless protocol, which means that each unit of data is individually addressed and routed from the source device to the target device, and the target does not send an acknowledgement back to the source. That’s where protocols such as the Transmission Control Protocol (TCP) come in. TCP is used in conjunction with IP in order to maintain a connection between the sender and the target and to ensure packet order.

For example, when an email is sent over TCP, a connection is established and a 3-way handshake is made. First, the source send an SYN “initial request” packet to the target server in order to start the dialogue. Then the target server then sends a SYN-ACK packet to agree to the process. Lastly, the source sends an ACK packet to the target to confirm the process, after which the message contents can be sent. The email message is ultimately broken down into packets before each packet is sent out into the Internet, where it traverses a series of gateways before arriving at the target device where the group of packets are reassembled by TCP into the original contents of the email.



The primary version of IP used on the Internet today is Internet Protocol Version 4 (IPv4). Due to size constraints with the total number of possible addresses in IPv4, a newer protocol was developed. The newer protocol is called IPv6 and it makes many more addresses available and is increasing in adoption.

UDP/IP

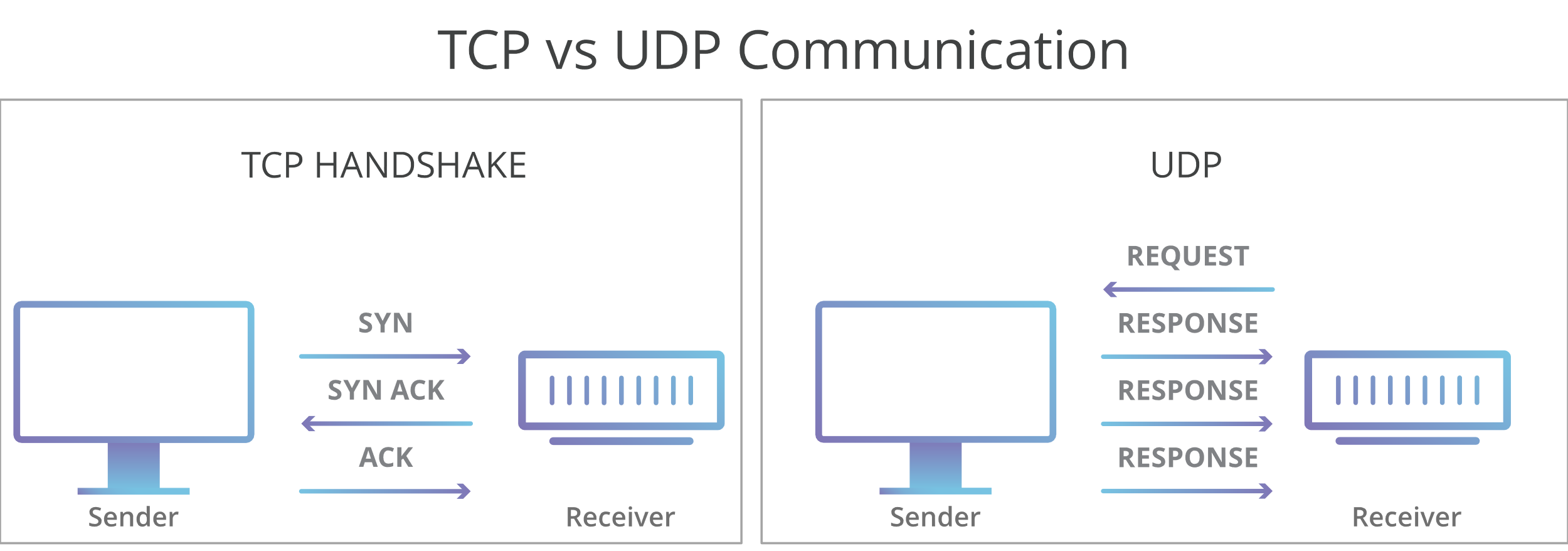
Learning Objectives

After reading this article you will be able to:

* Define UDP/IP
* Describe some use cases for UDP

## What is User Datagram Protocol (UDP/IP)?

UDP is a communication protocol used across the Internet for especially time-sensitive transmissions such as video playback or [DNS](https://www.cloudflare.com/learning/dns/what-is-dns/) lookups. It speeds up communications by not requiring what’s known as a “handshake”, allowing data to be transferred before the receiving party agrees to the communication. This allows the protocol to operate very quickly, and also creates an opening for exploitation.



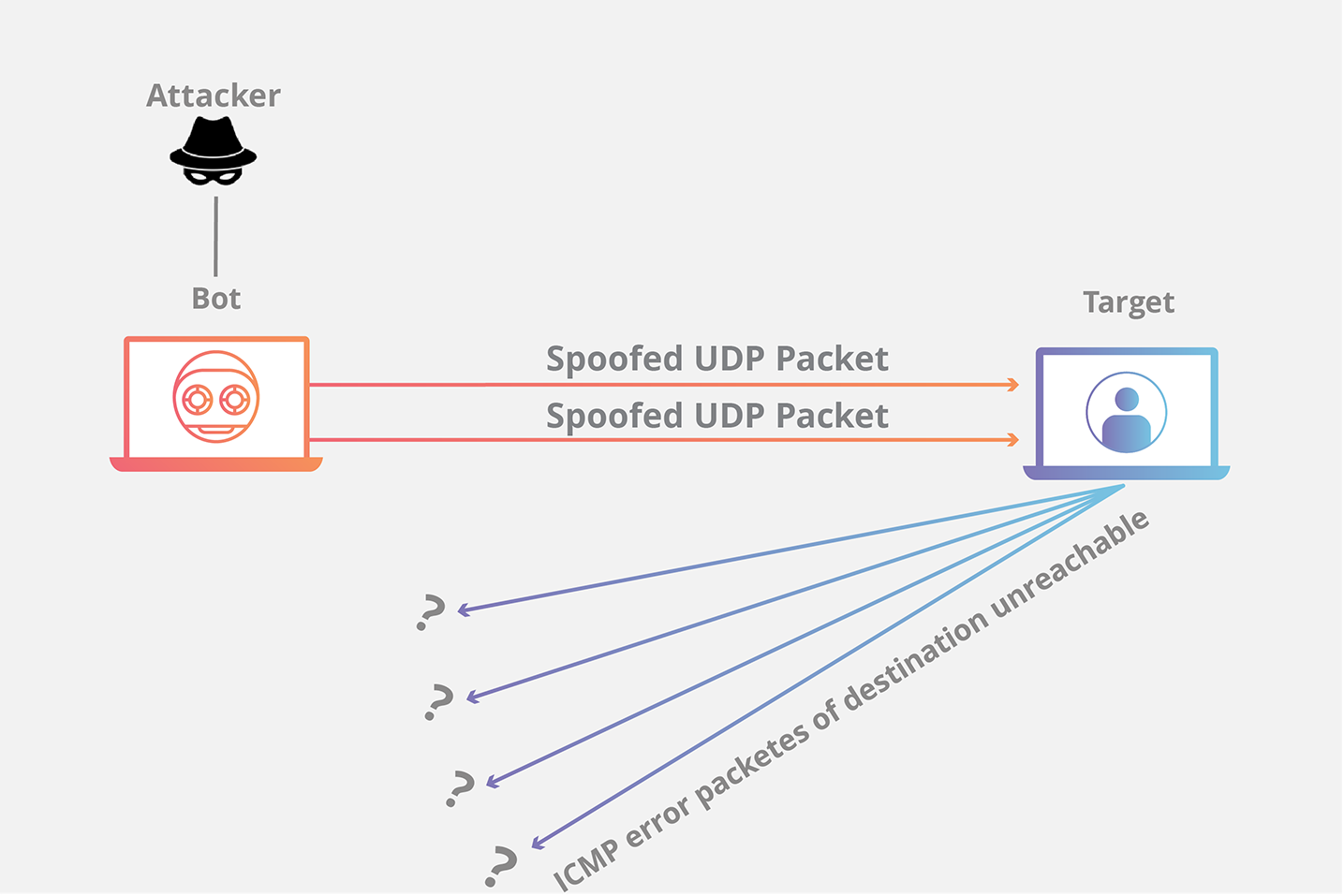
A [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) connection, which is used commonly used for loading web page content, requires a handshake in which the receiver agrees to the communication before the data is sent. UDP will send data without confirmation, even if the request is fraudulent.

UDP doesn’t have the error checking and ordering functionality of TCP and is best utilized when error checking is not needed and speed is important. This built-in lack of reliability is why UDP is sometimes referred to as ‘Unreliable Datagram Protocol’.

Introduced in 1980, UDP is among the oldest network protocols still in use. Applications that utilize UDP must be able to tolerate errors, loss, and duplication. While this sounds less than ideal, there are several applications where a faster and less reliable protocol is the best choice.

## What Kind Of Services Rely On UDP?

UDP is commonly used in time-sensitive communications where occasionally dropping packets is better than waiting. Voice and video traffic are sent using this protocol because they are both time-sensitive and designed to handle some level of loss. For example VOIP (voice over IP), which is used by many internet-based telephone services, operates over UDP. This is because a staticy phone conversation is preferable to one that is crystal clear but heavily delayed. This also makes UDP the ideal protocol for online gaming. Similarly, because DNS and NTP servers both need to be fast and efficient, they operate though UDP. Volumetric [DDoS attacks](https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/) including [DNS amplification](https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/) and [NTP amplification](https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/) make use of vulnerable instances of these servers with the aim of flooding a target with UDP traffic.



ICMP

Learning Objectives

After reading this article you will be able to:

* Define the ICMP
* Describe how ping and traceroute work

## What is the Internet Control Message Protocol (ICMP)?

The Internet Control Message Protocol is an internet layer protocol used by network devices to diagnose network communication issues. ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner. Commonly, the ICMP protocol is used on network devices, such as routers.

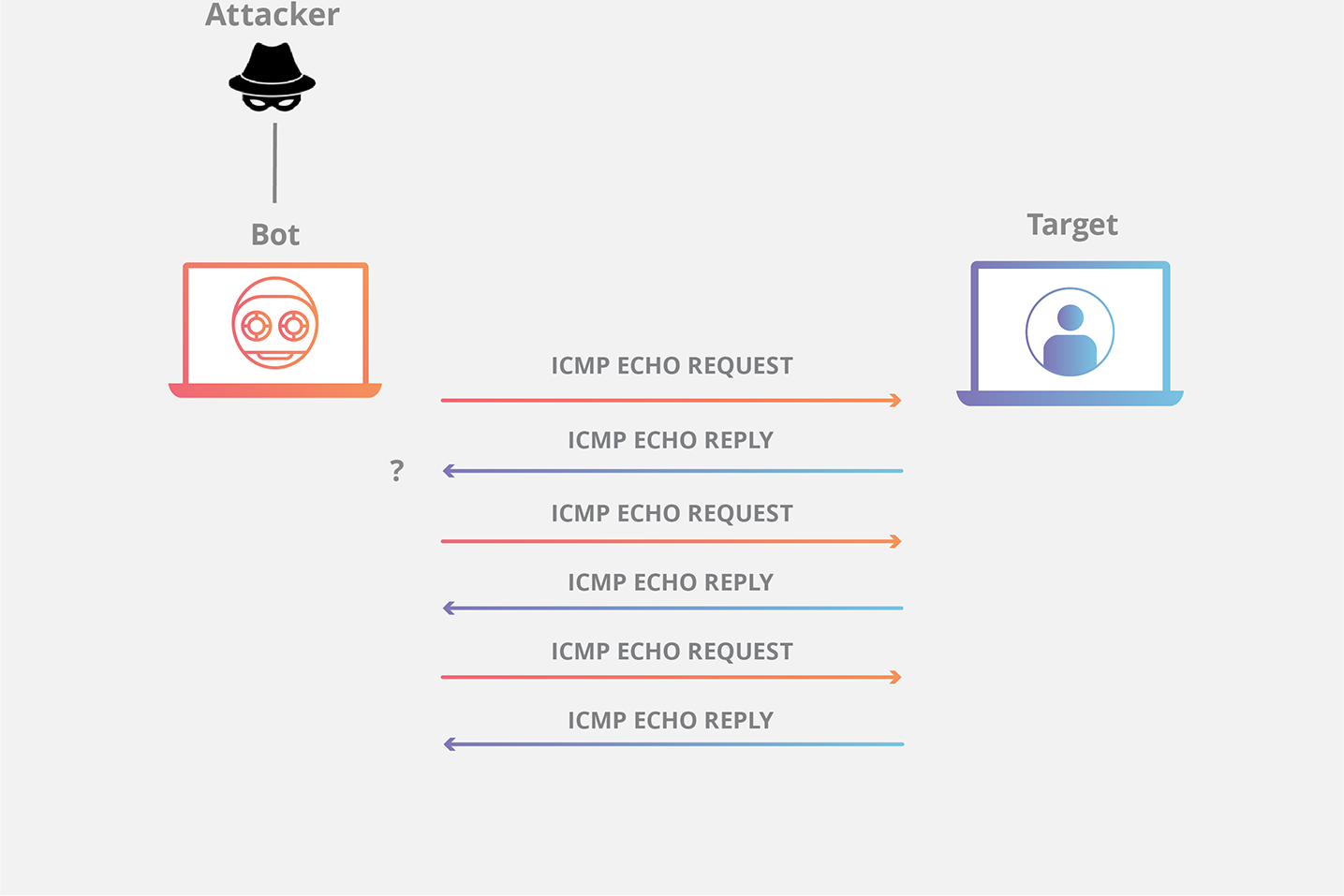
## What Is ICMP Used For?

The primary purpose of ICMP is for error reporting. When two devices connect over the Internet, the ICMP generates errors to share with the sending device in the event that any of the data did not get to its intended destination.

A secondary use of ICMP protocol is to perform network diagnostics; the commonly used terminal utilities traceroute and ping both operate using ICMP. The traceroute utility is used to display the routing path between two Internet devices. The routing path is the actual physical path of connected routers that a request must pass through before it reaches its destination. The journey between one router and another is known as a ‘hop’, and a traceroute also reports the time required for each hop along the way. This can be useful for determining sources of network delay.

The ping utility is a simplified version of traceroute. A ping will test the speed of the connection between two devices and report exactly how long it takes a packet of data to reach its destination and come back to the sender’s device. Although ping does not provide data about routing or hops, it is still a very useful metric for gauging the latency between two devices. The ICMP echo-request and echo-reply messages are commonly used for the purpose of performing a ping. Unfortunately network attacks can exploit this process, creating means of disruption such as the [ICMP Flood Attack](https://www.cloudflare.com/learning/ddos/ping-icmp-flood-ddos-attack/) and the [Ping of Death](https://www.cloudflare.com/learning/ddos/ping-of-death-ddos-attack/) attack.

ICMP Flood Attack:



DNS

Learning Objectives

After reading this article you will be able to:

* Define DNS
* Understand how a DNS request works
* Understand how DDoS attackers target DNS vulnerabilities

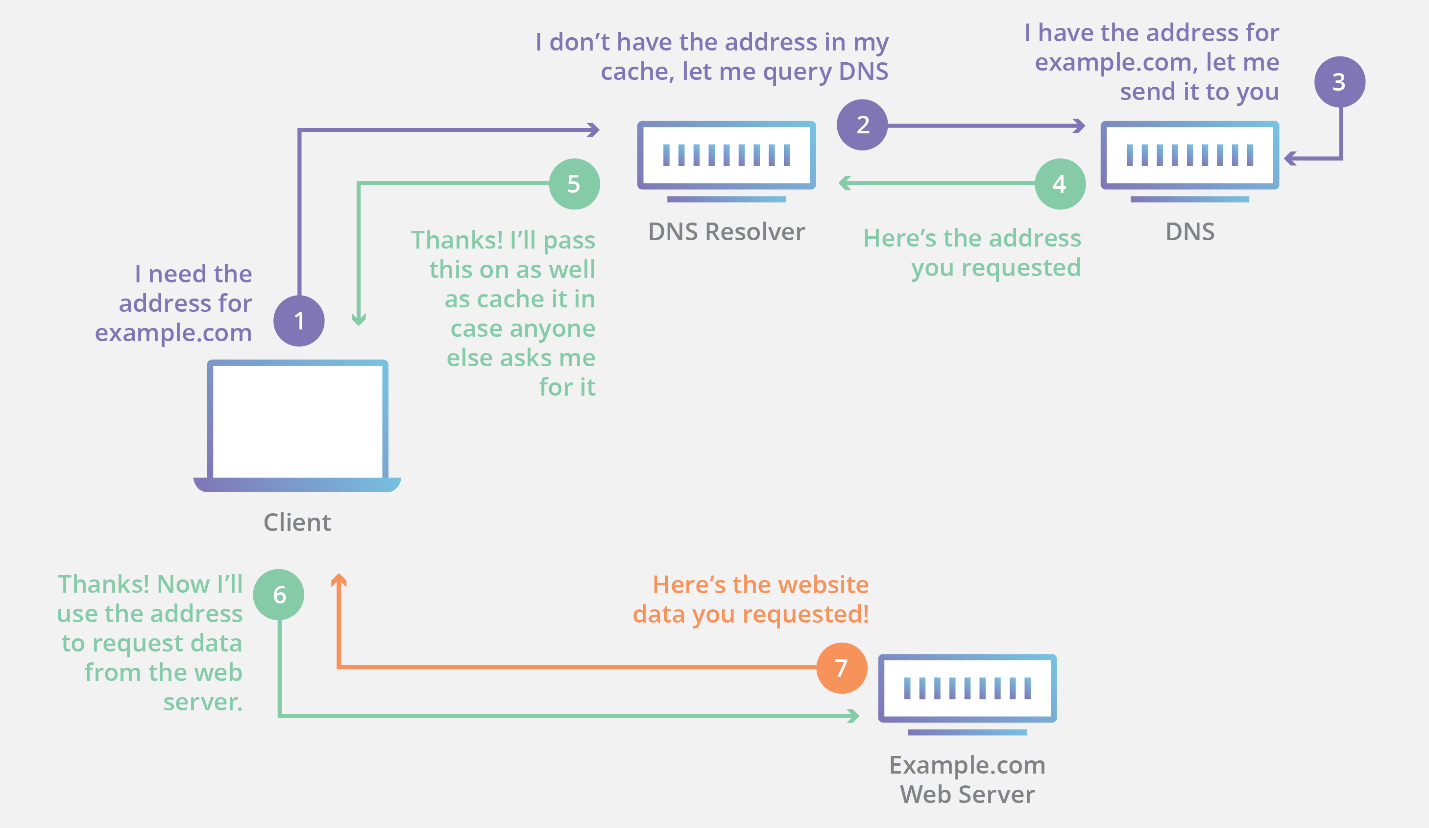
What is DNS?

DNS is often referred to as the phonebook of the internet, when a user types a web address into their browser, DNS is what connects that user with the web site they are seeking. DNS stands for Domain Name System, and the DNS maintains a directory of every website on the Internet.

A computer can only find a website using it’s [IP address](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/), which is a long, punctuated string of numbers, such as 192.168.1.1 in the older IPv4 format, or 2400:cb00:2048:1::c629:d7a2 in the new IPv6 . These addresses can be hard for humans to remember, and on top of that, the IP addresses for some websites are dynamic and can change periodically. DNS makes it easier for people to access websites by letting them use human-friendly web addresses, also known as URLs.

For example, a current IPv6 IP address for Cloudflare.com is 2400:cb00:2048:1::c629:d7a2. Instead of memorizing that address, a user can type ‘www.cloudflare.com’ into their browser. When that happens, the browser sends out a request to DNS, and DNS returns a response telling the browser the IP address of that website, and the browser then sends a request to that IP address which responds with the website’s data.

How does a DNS request work?



[DNS servers](https://www.cloudflare.com/learning/dns/dns-server-types/) are set up in a distributed hierarchy, meaning the data is spread out over several computers. When a client makes a DNS request, the request is handled by a recursive resolver, which is a DNS server that starts a series of communications with other DNS servers until it finds the requested IP address, returning it to the client. Recursive resolvers can also cache DNS records, making frequently accessed records more readily available.

DNS and DDoS Attacks

There are two popular [DDoS attacks](https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/) that utilize DNS servers: DNS amplification attacks and DNS flood attacks.

* DNS amplification attacks are reflection-based DDoS attacks where the attacker sends [spoofed](https://www.cloudflare.com/learning/ddos/glossary/ip-spoofing/) look-up requests to an open DNS server, and the server then sends the responses back to a targeted victim. The attack is amplified because the request data sent by the attacker is smaller than the response data received by the victim. [Learn more about DNS amplification attacks.](https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/)
* In a DNS flood attack, the attackers attempt to overwhelm the DNS servers for a particular zone in an attempt to disrupt legitimate traffic to that zone. This type of attack is generally done by using a [botnet](https://www.cloudflare.com/learning/ddos/what-is-a-ddos-botnet/) to overwhelm a DNS resolver with lookup requests. [Learn more about DNS flood attacks.](https://www.cloudflare.com/learning/ddos/dns-flood-ddos-attack/)

What else does the Domain Name System do?

The DNS also defines the DNS Protocol, which is a detailed specification of communication exchanges and data structures used in the DNS. This falls under the Internet Protocol Suite ([TCP/IP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/)). Additionally, The DNS maintains a master [Blackhole](https://www.cloudflare.com/learning/ddos/glossary/ddos-blackhole-routing/) List of IP addresses known for sending out spam email. Mail servers can be configured based on this list to flag or reject messages suspected to be spam.

HTTP

Learning Objectives

After reading this article you will be able to:

* Define HTTP
* Describe the anatomy of HTTP requests and responses
* Understand how DDoS attacks can be launched over HTTP

## What is HTTP?

The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, and is used to load web pages using hypertext links. HTTP is an [application layer](https://www.cloudflare.com/learning/ddos/application-layer-ddos-attack/) protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack. A typical flow over HTTP involves a client machine making a request to a server, which then sends a response message.

## What’s in an HTTP request?

An HTTP request is the way internet communications platforms such as web browsers ask for the information they need to load a website.

Each HTTP request made across the Internet carries with it a series of encoded data that carries different types of information. A typical HTTP request contains:

1. HTTP version type
2. a URL
3. an HTTP method
4. HTTP request headers
5. Optional HTTP body.

Let’s explore in greater depth how these requests work, and how the contents of a request can be used to share information.

#### What’s an HTTP method?

An HTTP method, sometimes referred to as an HTTP verb, indicates the action that the HTTP request expects from the queried server. For example, two of the most common HTTP methods are ‘GET’ and ‘POST’; a ‘GET’ request expects information back in return (usually in the form of a website), while a ‘POST’ request typically indicates that the client is submitting information to the web server (such as form information, e.g. a submitted username and password).

#### What are HTTP request headers?

HTTP headers contain text information stored in key-value pairs, and they are included in every HTTP request (and response, more on that later). These headers communicate core information, such as what browser the client is using what data is being requested.

Example of HTTP request headers from Google Chrome's network tab:

#### What’s in an HTTP request body?

The body of a request is the part that contains the ‘body’ of information the request is transferring. The body of an HTTP request contains any information being submitted to the web server, such as a username and password, or any other data entered into a form.

## What’s in an HTTP response?

An HTTP response is what web clients (often browsers) receive from an Internet server in answer to an HTTP request. These responses communicate valuable information based on what was asked for in the HTTP request.

A typical HTTP response contains:

1. an HTTP status code
2. HTTP response headers
3. optional HTTP body

Let's break these down:

#### What’s an HTTP status code?

HTTP status codes are 3-digit codes most often used to indicate whether an HTTP request has been successfully completed. Status codes are broken into the following 5 blocks:

* 1. 1xx Informational
  2. 2xx Success
  3. 3xx Redirection
  4. 4xx Client Error
  5. 5xx Server Error

The “xx” refers to different numbers between 00 and 99.

Status codes starting with the number ‘2’ indicate a success. For example, after a client requests a web page, the most commonly seen responses have a status code of ‘200 OK’, indicating that the request was properly completed.

If the response starts with a ‘4’ or a ‘5’ that means there was an error and the webpage will not be displayed. A status code that begins with a ‘4’ indicates a client-side error (It’s very common to encounter a ‘404 NOT FOUND’ status code when making a typo in a URL). A status code beginning in ‘5’ means something went wrong on the server side. Status codes can also begin with a ‘1’ or a ‘3’, which indicate an informational response and a redirect, respectively.

#### What are HTTP response headers?

Much like an HTTP request, an HTTP response comes with headers that convey important information such as the language and format of the data being sent in the response body.

Example of HTTP response headers from Google Chrome's network tab:

#### 

#### What’s in an HTTP response body?

Successful HTTP responses to ‘GET’ requests generally have a body which contains the requested information. In most web requests, this is HTML data which a web browser will translate into a web page.

## Can DDoS Attacks Be Launched Over HTTP?

Keep in mind that HTTP is a “stateless” protocol, which means that each command runs independent of any other command. In the original spec, HTTP requests each created and closed a [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/) connection. In newer versions of the HTTP protocol (HTTP 1.1 and above), persistent connection allows for multiple HTTP requests to pass over a persistent TCP connection, improving resource consumption. In the context of [DoS](https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/) or [DDoS attacks](https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/), HTTP requests in large quantities can be used to mount an attack on a target device, and are considered part of application layer attacks or [layer 7](https://www.cloudflare.com/learning/ddos/what-is-layer-7/) attacks.