# Difference between http1.1 &http2

## What is HTTP?

Hypertext Transfer Protocol (HTTP) is an application protocol that is, currently, the foundation of data communication for the World Wide Web.

**HTTP** is based on the Client/Server model. Client/Server model can be explained as two computers, Client (receiver of service) and Server (provider of service) that are communicating via requests and responses.

A simple and abstract example would be a restaurant guest and a waiter. The guest (**Client**) asks (**sends request**) waiter (**Server**) for a meal, then the waiter gets the meal from the restaurant chef (**your application logic**) and brings the meal to the guest.

The first usable version of HTTP was created in 1997. Because it went through several stages of development, this first version of HTTP was called HTTP/1.1. This version is still in use on the web. In 2015, a new version of HTTP called **HTTP/2** was created.

HTTP/2 solves several problems that the creators of HTTP/1.1 did not anticipate. In particular, HTTP/2 is much faster and more efficient than HTTP/1.1. One of the ways in which HTTP/2 is faster is in how it prioritizes content during the loading process.

## What is Prioritization?

In the context of web performance, prioritization refers to the order in which pieces of content are loaded. Suppose a user visits a new website and navigates to an article. Should the top of the article load first? Should the text of the article load first? Should the banner ads loads first?

Prioritization affects a webpage’s load time. For example, certain resources, like JavaScript files, may block the rest of the page from loading if they have to load first. More of the page can load at once if these render-blocking resources load last.

In addition, the order in which these pages resources load affects how the user perceives page load time. If only behind-the-scenes content (like a CSS file) or content the user can’t see immediately (like banner ads at the bottom of the page) loads first, the user will think the page is not loading at all. If the content that’s most important to the user loads first, such as the image at the top of the page, then the user will perceive the page as loading faster.

## How does prioritization in HTTP/2 affect performance?

In HTTP/2, developers have hands-on, detailed control over prioritization. This allows them to maximize perceived and actual page load speed to a degree that was not possible in HTTP/1.1.

HTTP/2 offers a feature called weighted prioritization. This allows developers to decide which page resources will load first, every time. In HTTP/2, when a client makes a request for a webpage, the server sends several streams of data to the client at once, instead of sending one thing after another. This method of data delivery is known as multiplexing. Developers can assign each of these data streams a different weighted value, and the value tells the client data streams to render first.

Imagine that Alice wants to read a novel that her friend Bob wrote, but both Alice and Bob only communicate through the regular mail. Alice sends a letter to Bob and asks Bob to send her his novel. Bob decides to send the novel

**HTTP/1.1 – style:** He mails one chapter at a time, and he only mails the next chapter after receiving a reply letter from Alice confirming that she received the previous chapter. Using this method of content delivery, it takes Alice many weeks to read Bob’s novel.

Now imagine that Bob decides to send Alice his novel

**HTTP/2 –style:** In this case, he sends each chapter of the novel separately (to stay within the postal service’s size limits) but all at the same time. He also numbers each chapter: Chapter 1, Chapter 2, etc. Now, Alice receives the novel all at once and can assemble it in the correct order on her own time. If a chapter is missing, she may send a quick reply asking for that specific chapter, but otherwise the process is complete, and Alice can read the novel in just a few days.

HTTP/2, data is send all at once, much like Bob when he sends Alice multiple chapter at once. And just like Bob, developers get to number the chapters in HTTP/2. They can decide if the text of a webpage loads first, or the CSS files, or the JavaScript, or whatever they feel is most important for the user experience.

## What are the other difference between HTTP/2 and HTTP/1.1 that impact performance?

### Multiplexing:

HTTP/1.1 loads resources one after the other, so if one resource cannot be loaded, it blocks all the other resources behind it. In contract, HTTP/2 is able to use a single TCP connection to send multiple streams of data at once so that no one resource blocks any other resources. HTTP/2 does this by splitting data into binary-code messages and numbering these messages so that the client knows which streams each binary message belongs to.

### Server push:

Typically, a server only serves content to a client ask for it. However, this approach is not always practical for modern web pages, which often involves several dozen separate resources that the client must request. HTTP/2 solves this problem by allowing a server to “push” content to a client asks for it. The server also sends a message letting the client know what pushed content to expect –like if Bob had send Alice a Table of Contents of his novel before sending the whole thing.

### Header compression:

Small files load more quickly than large ones. To speed up web performance, both HTTP/1.1 & HTTP/2 compress HTTP message to make them smaller. However, HTTP/2 uses a more advanced compression method called HPACK that eliminates redundant information in HTTP header packets. This eliminates a few bytes from every HTTP packet. Given the volume of HTTP packets involves in loading even a single webpage, those bytes add up quickly, resulting in faster loading.