# Problem

* **Explain briefly what happens when you hit a URL? explain DNS lookup**

URL stands for Uniform Resource Locator. URL is the address of the website which you can find in the address bar of your web browser. It is a reference to a resource on the internet, be it images, hypertext pages, audio/video files, etc.

**What is** DNS **:**

DNS is short for Domain Name System. Like a phonebook, DNS maintains and maps the name of the website, i.e. URL, and the particular IP address it links to. Every URL on the internet has a unique IP address which is of the computer which hosts the server of the website requested.

**Steps for what happens when we enter a URL :**

1. Browser checks cache for DNS entry to find the corresponding IP Address of website.  
   It looks for the following cache. If not found in one, then continues checking to the next until found.
   * Browser Cache
   * Operating Systems Cache
   * Router Cache
   * ISP Cache
2. If not found in the cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find the IP address of the server that hosts the domain name.  
   The requests are sent using small data packets that contain information content of the request and the IP address it is destined for.
3. The browser initiates a TCP (Transfer Control Protocol) connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
4. The browser sends an HTTP request to the webserver. GET or POST request.
5. The server on the host computer handles that request and sends back a response. It assembles a response in some formats like JSON, XML, and HTML.
6. The server sends out an HTTP response along with the status of the response.
7. Browser displays HTML content
8. Finally, Done.

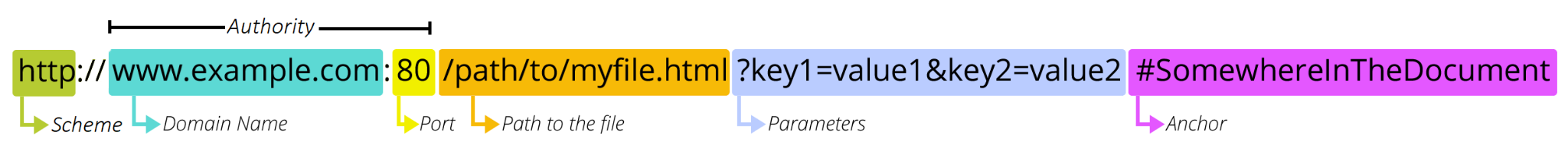
* **What is a URL's full form? Explain what a URL is and the four parts it consists of The protocol in use The hostname of the server The location of the file The arguments to the file**

*URL* is one of the key concepts of the Web. It is the mechanism used by browsers to retrieve any published resource on the web.

URL stands for *Uniform Resource Locator*. A URL is nothing more than the address of a given unique resource on the Web. In theory, each valid URL points to a unique resource. Such resources can be an HTML page, a CSS document, an image, etc. In practice, there are some exceptions, the most common being a URL pointing to a resource that no longer exists or that has moved. As the resource represented by the URL and the URL itself is handled by the Web server, it is up to the owner of the webserver to carefully manage that resource and its associated URL.

Any of those URLs can be typed into your browser's address bar to tell it to load the associated page (resource).

A URL is composed of different parts, some mandatory and others optional. The most important parts are highlighted on the URL below (details are provided in the following sections):



* **What is HTTP protocol?**

HTTP is an extensible protocol that has evolved over time. It is an application layer protocol that is sent over TCP, though any reliable transport protocol could theoretically be used. Due to its extensibility, it is used to not only fetch hypertext documents but also images and videos or to post content to servers, like with HTML form results. HTTP can also be used to fetch parts of documents to update Web pages on demand.

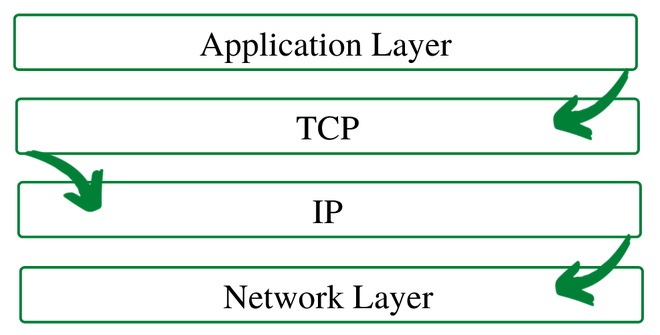
HTTP is a protocol for fetching resources such as HTML documents. It is the foundation of any data exchange on the Web and it is a client-server protocol, which means requests are initiated by the recipient, usually the Web browser. A complete document is reconstructed from the different sub-documents fetched, for instance, text, layout description, images, videos, scripts, and more.



Clients and servers communicate by exchanging individual messages (as opposed to a stream of data). The messages sent by the client, usually a Web browser, are called *requests,* and the messages sent by the server as an answer are called *responses*.

* **What is TCP Protocol?**

TCP (Transmission Control Protocol) is one of the main protocols of the Internet protocol suite. It lies between the Application and Network Layers which are used in providing reliable delivery services. It is a connection-oriented protocol for communications that helps in the exchange of messages between the different devices over a network.



* **Explain all the different HTTP methods?**

# **HTTP request methods**

HTTP defines a set of request methods to indicate the desired action to be performed for a given resource. Each of them implements a different semantic, but some common features are shared by a group of them: e.g. a request method can be safe, idempotent, or cacheable.

GET

The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.

HEAD

The HEAD method asks for a response identical to a GET request, but without the response body.

POST

The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server.

PUT

The PUT method replaces all current representations of the target resource with the request payload.

DELETE

The DELETE method deletes the specified resource.

CONNECT

The CONNECT method establishes a tunnel to the server identified by the target resource.

OPTIONS

The OPTIONS method describes the communication options for the target resource.

TRACE

The TRACE method performs a message loop-back test along the path to the target resource.

PATCH

The PATCH method applies partial modifications to a resource.

* **What are HTTP headers?**

The **HTTP headers** are used to pass additional information between the clients and the server through the **request** and **response** header. All the headers are case-insensitive, headers fields are separated by a colon, and key-value pairs are in clear-text string format. The end of the header section is denoted by an empty field header. There are a few header fields that can contain the comments. And a few headers can contain quality(q) key-value pairs that are separated by an equal sign.

**There are four kinds of headers context-wise:**

* **General Header:** This type of header is applied to Request and Response headers both but without affecting the database body.
* **Request Header:** This type of header contains information about the fetched request by the client.
* **Response Header:** This type of header contains the location of the source that has been requested by the client.
* **Entity Header:** This type of header contains the information about the body of the resources like MIME type, Content-length.
* **What are some HTTP response codes? what does it mean? 2xx, 3xx, 4xx, 5xx**

The Status-Code element is a server response, is a 3-digit integer where the first digit of the Status-Code defines the class of response and the last two digits do not have any categorization role. There are 5 values for the first digit:

| **S.N.** | **Code and Description** |
| --- | --- |
| 1 | 1xx: Informational  It means the request has been received and the process is continuing. |
| 2 | 2xx: Success  It means the action was successfully received, understood, and accepted. |
| 3 | 3xx: Redirection  It means further action must be taken in order to complete the request. |
| 4 | 4xx: Client Error  It means the request contains incorrect syntax or cannot be fulfilled. |
| 5 | 5xx: Server Error  It means the server failed to fulfill an apparently valid request. |

HTTP status codes are extensible and HTTP applications are not required to understand the meaning of all the registered status codes. Given below is a list of all the status codes.

* **What does cache control on HTTP response mean?**

The Cache-Control HTTP header field holds *directives* ching in browsers and shared caches (e.g. Proxies, CDNs).

* **What is polling?**

**Polling** is continuous, repetitive connections made to a computer or device, to verify its proper operation. This process could occur thousands+ of times every minute.

'polling' is the continuous checking of other programs or devices by one program or device to see what state they are in, usually to see whether they are still connected or want to communicate.

* **What is long polling?**

Long polling is the simplest way of having a persistent connection with the server, that doesn’t use any specific protocol like WebSocket or Server Side Events. Being very easy to implement, it’s also good enough in a lot of cases.

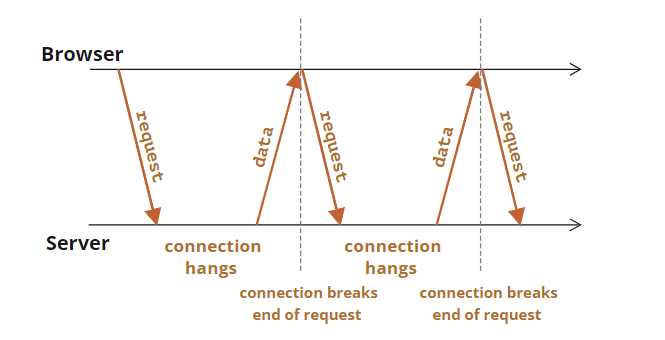
So-called “long polling” is a much better way to poll the server.

It’s also very easy to implement and delivers messages without delays.

The flow:

1. A request is sent to the server.
2. The server doesn’t close the connection until it has a message to send.
3. When a message appears – the server responds to the request with it.
4. The browser makes a new request immediately.

The situation when the browser sent a request and has a pending connection with the server is standard for this method. Only when a message is delivered, the connection is reestablished.



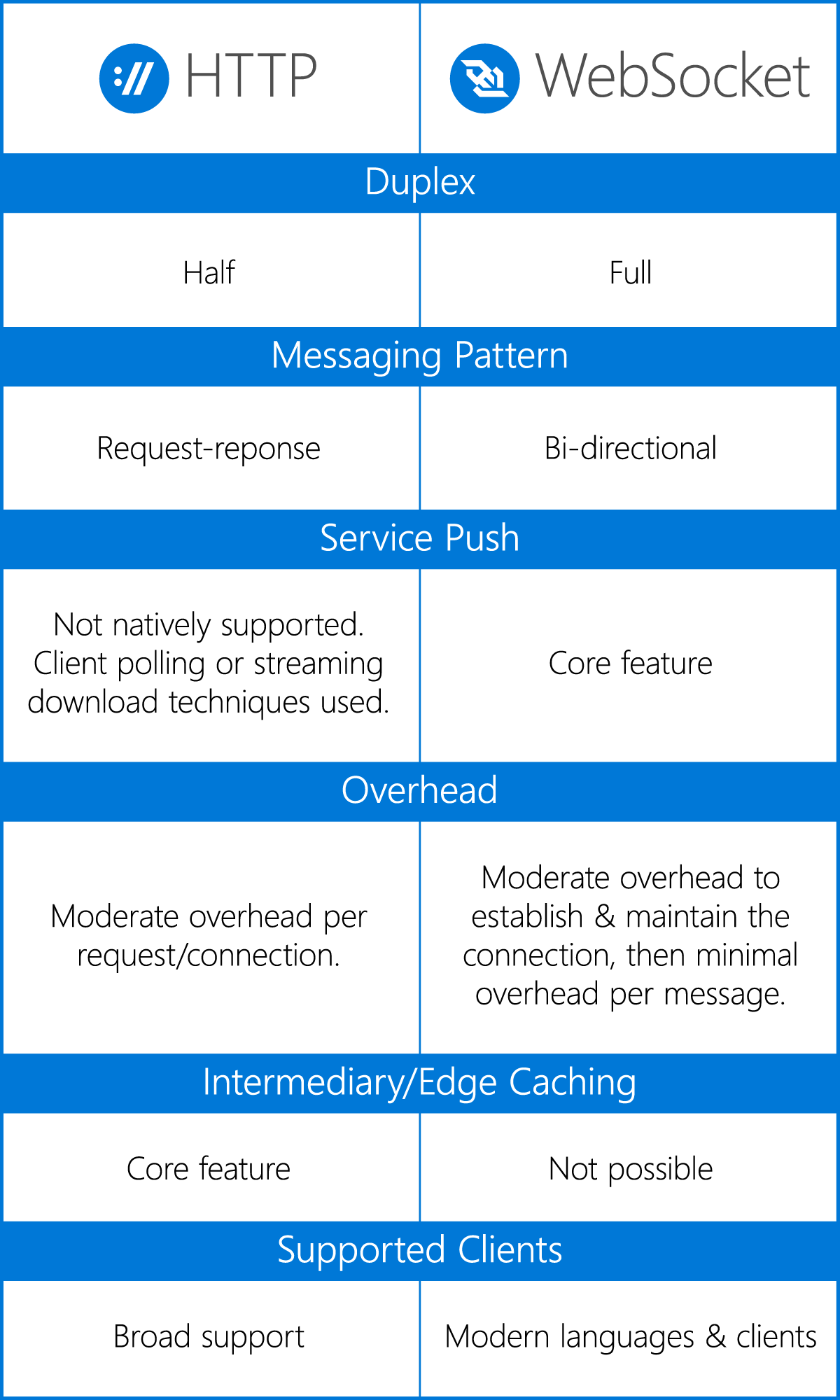
If the connection is lost, because of, say, a network error, the browser immediately sends a new request.

A sketch of the client-side subscribe function that makes long requests:

* **What are web sockets?**

The WebSocket API is an advanced technology that makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

* **How are web sockets different from HTTP?**



* **What is HTTPS?**

Hypertext Transfer Protocol Secure is an extension of the Hypertext Transfer Protocol. It is used for secure communication over a computer network and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security or, formerly, Secure Sockets Layer.

* **What is Cross-Origin Resource Sharing? ( CORS ) Why do we need it?**

Cross-Origin Resource Sharing (CORS) is an HTTP-header-based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources. CORS also relies on a mechanism by which browsers make a "preflight" request to the server hosting the cross-origin resource, in order to check that the server will permit the actual request. In that preflight, the browser sends headers that indicate the HTTP method and headers that will be used in the actual request.

* What does the Access-Control-Allow-Origin header do?
* What is clickjacking? How do you fix it?
* What are some performance metrics for your website?
* What does the following term mean?
  + Time to first byte,
  + Page load time
* What do CDN or Content Delivery Networks do? When do you need a CDN?
* What is the difference between Client-Side Rendering and Server Side Rendering?
* What is Progressive Rendering
* What is the difference between Preloading and Prefetching resources?
* What are service workers?