Task No.1

Q.1) Difference between HTTP1.1 vs HTTP2.

|  |  |  |
| --- | --- | --- |
| Parameters | HTTP 1.1 | HTTP 2 |
| Key Features | It supports connection reuse i.e. for every TCP connection there could be multiple requests and responses, and pipelining where the client can request several resources from the server at once. However, pipelining was hard to implement due to issues such as head-of-line blocking and was not a feasible solution. | Uses multiplexing, where over a single TCP connection resources to be delivered are interleaved and arrive at the client almost at the same time. It is done using streams which can be prioritized, can have dependencies and individual flow control. It also provides a feature called server push that allows the server to send data that the client will need but has not yet requested. |
| Status Code | Introduces a warning header field to carry additional information about the status of a message. Can define 24 status codes, error reporting is quicker and more efficient. | Underlying semantics of HTTP such as headers, status codes remains the same. |
| Authentication Mechanism | It is relatively secure since it uses digest authentication, NTLM authentication. | Security concerns from previous versions will continue to be seen in HTTP/2. However, it is better equipped to deal with them due to new TLS features like connection error of type Inadequate\_Security. |
| Caching | Expands on the caching support by using additional headers like cache-control, conditional headers like If-Match and by using entity tags. | HTTP/2 does not change much in terms of caching. With the server push feature if the client finds the resources are already present in the cache, it can cancel the pushed stream. |
| Performance Optimization | Spriting, concatenating, inlining, domain sharding are some of the optimizations used as a workaround to the ‘six connections per host’ rule | Removes the need for unnecessary optimization hacks |
| Security | SSL is not required but recommended. Digest authentication used in HTTP1.1 is an improvement over HTTP1.0. HTTPS uses SSL/TLS for secure encrypted communication. | Though security is still not mandatory, it is mostly encrypted (though it is not enforced) since almost all clients require traffic to be encrypted. It also has some minimum standards, such as minimum key size for encryption. TLS 1.2 etc. |

Q.2) http version history.

1.HTTP/0.9

-Year of release 1991.

-It is extremely simple requests consist of a single line and start with the only possible method GET followed by the path to the resource.

-There were no status or error messages.

2.HTTP/1.0

-Year of release is 1996.

-Version information is now sent within each request.

-A status code is also sent at the beginning of the response.

-The HTTP headers has been introduced.

-The transfer of other documents than plain HTML files has been added.

3.HTTP/1.1

-Year of release is 1997.

-A connection can be reused.

-Pipelining has been added, allowing to send a second request before the answer for the first one is fully transmitted.

-Cache control mechanism have been introduced.

4.HTTP/2

-Year of release is 2015.

-It is a binary protocol rather than text.

-It is a multiplexed protocol. Parallel requests can be handled over the same connection, removing the order and blocking issues of the HTTTP/1.1 protocol.

-It compresses headers. As these are often similar among a set of requests, this removes duplication and overhead of data transmitted.

Q.3) List 5 difference between Browser JS(console) vs Nodejs.

1. In browser “window” is a predefined global object which has functions and attributes, whereas Nodejs doesn’t have it.

2. In browser “location” is another predefined object, whereas Nodejs doesn’t have it.

3. In browser “require” is not predefined object, whereas Nodejs has it.

4. In browser module is not required, where as in Nodejs you have to keep your code inside the module.

5. In browser “document” is a predefined object, whereas Nodejs doesn’t have it.

Q.4) What happens when you type a URL in the address bar in the browse?

1. Browser checks cache for DNS entry to find the corresponding [IP address](https://www.geeksforgeeks.org/introduction-of-classful-ip-addressing/) of website.  
   It looks for 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 cache, ISP’s (Internet Service Provider) DNS server initiates a DNS query to find IP address of server that hosts the domain name.  
   The requests are sent using small data packets that contain information content of request and IP address it is destined for.
3. Browser initiates a [TCP (Transfer Control Protocol)](https://www.geeksforgeeks.org/tcp-and-udp-in-transport-layer/) connection with the server using synchronize(SYN) and acknowledge(ACK) messages.
4. Browser sends an [HTTP](https://www.geeksforgeeks.org/http-non-persistent-persistent-connection/) request to the web server. GET or POST request.
5. Server on the host computer handles that request and sends back a response. It assembles a response in some format like JSON, [XML](https://www.geeksforgeeks.org/xml-basics/) and HTML.
6. Server sends out an HTTP response along with the status of response.
7. Browser displays [HTML](https://www.geeksforgeeks.org/html-tutorials/) content