TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

# Advancement of HTTP and Security Issues Over It

Nitya Nand Dwivedi[1], Dharmendra Singh[2]
[1,2]Babu Sunder Singh Institute of Technology and Management, Uttar Pradesh, India
[1]nityananddwivedi29@gmail.com, [2]dharmendra11singh@gmail.com

KEYWORD

HTTP/2,
SPDY,
HTTP/1,
Cookies,
Encryption of Cookies.

ABSTRACT

*Now days so many organizations are focusing that how to improve http and security related problem over it. The new protocol, HTTP / 2, has most popular in the web industry as its typical information was confirmed and accepted earlier few years. Many of its technical structures are derived from the Google application-based protocol, HTTP / 2 solves many bugs and variations of HTTP / 1.1, improving web performance during page loading times. Expected. HTTP / 2 introduces topic compression, which allows multiple simultaneous exchanges on the same connection, allowing better utilization of network resources and reducing latency. We will also introduce one-sided submissions from the server to the client. Flow and priority control can help you use more live streaming. Flow control helps ensure that only data received from the recipient is sent. Priority allows the index of limited resources to the most significant load first. HTTP / 2 adds a novel communication method that allows the server that may be send feedback to the client.*
*This research article shows that this functionality learns the impact of HTTP / 2 performance on standard websites, and while using HTTP / 2 improves page load speed and efficiency, increasing cookie usage makes it less secure. Therefore, you need to prevent this and ensure that the cookies you use are securely stored in your system and encrypted in some way. You can allow encryption locally, but you can also re-use the website upload server. In my research I used encryption via Node.js and the XAMP server.*

## 1. Introduction

The term hypertext was developed by Ted Nelson in 1965 in the Xanadu Project, encouraged through Vannevar Bush's 1930 vision of microfilm-based data and the "memex" management framework featured in his 1945 "As We may" story. Tim Berners-Lee and his group at CERN are said to have created unique HTTP and HTML and related web server and online internet browser innovation. Berners-Lee previously proposed an "internet" project in 1989 - now acknowledged as the World Wide Web. The principal variant of the convention had just a single technique, to be specific GET, which could demand a page on the server. [5] Feedback from the server has forever been a HTML page. The primary composed variant of HTTP was HTTP V0.9 (1991). Dave

**Corresponding Author: Nitya Nand Dwivedi,** Babu Sunder Singh Institute of Technology and Management, Uttar Pradesh, India
**Email:** nityananddwivedi29@gmail.com

Raggett drove the HTTP Working Group (HTTP WG) in 1995 and looked to grow the convention with broadened usefulness, upgraded discourse, rich meta data, coordinated and secure security convention by adding extra strategies and title fields. RFC 1945 formally sent off and perceived HTTP V1.0 in 1996.

### 1.1: HTTP/1.1:

HTTP WG wanted to distribute the novel principles in December 1995 and the past HTTP/1.1 help considering the overhauled RFC 2068 (called HTTP-NG) was quickly embraced. by significant program designers in mid-1996. In March 1996, the past HTTP/1.1 depended on Arena, Netscape 2.0, Netscape Navigator Gold 2.01, Mosaic 2.7, Lynx 2.5, and Internet Explorer 2.0. End client acknowledgment of new programs immediately sped up [4]. In March 1996, a web facilitating organization detailed that over 40% of Internet programs were consistent with HTTP 1.1. A similar web facilitating organization detailed that in June 1996, 65% of all programs getting to their servers were consistent with HTTP/1.1. The HTTP/1.1 norm as characterized in RFC 2068 was authoritatively delivered in January 1997. Redesigns and bring up-to-date to the HTTP/1.1 standard were given below RFC 2616 in June 1999.In 2007, the HTTP bis Working Group was established, in part, to review and clarify HTTP / 1.1 conditions. In June 2014, WG released a six-part updated version of RFC 2616:

• RFC 7230, HTTP / 1.1: Message syntax and Route
• RFC 7231, HTTP / 1.1: Semantics and content
• RFC 7232, HTTP / 1.1: Conditional applications
• RFC 7233, HTTP / 1.1: Grade Applications
• RFC 7234, HTTP / 1.1: Cache
• RFC 7235, HTTP / 1.1: Authentication

HTTP/2 is the up-and-coming age of HTTP convention, and accordingly, it ought to tackle past issues and bring further developed execution. With full testing, we are attempting to carry solid insights to this new adaptation of the convention [2]. To do this, we utilize a strategic methodology, considering the recently presented highlights: pressure, duplication, server push and need. Past exploration has demonstrated SPDY convention to be more noxious than HTTP on portable organizations. Likewise, in this work, we feature that HTTP/2 is adversely obstructed by parcel misfortune which is a component of portable organizations. We additionally checked with server drive once first to figure out their way of behaving. Since these techniques leave a ton of opportunity for novices, they have extraordinary power; and the present moment, they are not being taken advantage of delicately. With everything considered, we get down to earth data on the amount one can acquire by changing their assets to HTTP/2.

The advent of web services on rich web pages introduces complexity and page load times. In fact, in the late 1990s, web pages were just text. If you were lucky, it was a specific color. But today, we are dealing with an interactive web page that displays hundreds of images and uses many scripts. Each time you press a key, all mouse movements are recognized and affect the display of the web page. This behaviour is costly, especially because the traffic between the user and the server hosting the website is very difficult. Developed in the 1990s to provide links between browsers and web servers, Hypertext Transfer Protocol (HTTP) has sought to adapt to ever-changing web usage. This created multiple versions of the protocol (0.9, 1.0, 1.1). In 2009, Google introduced a test protocol called SPDY [6]. It retains HTTP semantics, but solves the HTTP / 1.1-line blocking issue. In 2012, the Internet Technology Commission [1] decided to investigate the details of the new HTTP version, HTTP / 2. The first HTTP / 2 framework in 2012 was based on the SPDY protocol. Since then, it has been very popular and has always been trying to reduce web page load times with HTTP / 2.

This task will help you better understand the actual HTTP / 2 protocol. Rather, the impact on this paper has tripled. First, configure your Chromium browser to disable the default TLS layer for a better comparison between HTTP / 2 and HTTP / 1.

A user interface base represented by browsers, networks, and mobile protocol cables on servers or "engines" this kind process applications and access several media. The HTTP (Hypertext Transfer Protocol) is under the World Wide Web. Of HTTP / 1.0, the hypertext transfer protocol is an application layer protocol with the light weight and speed required for shared hypermedia information systems. This is a common unconventional fixed protocol that can be used for many tasks, including: B. For name servers and distributed asset management systems by extending request methods (commands) without sending data.

## 1.2: SPDY Protocol (After HTTP/1.1):

Since July 2012, the SPDY Improvement Group has publicly stated its intention to investigate the outage (available as a web draft). The main HTTP / 2 system contains SPDY as the active structure for editing and modification. SPDY features are available in Chromium, Mozilla Firefox, Opera, Amazon Silk, Internet Explorer, and Safari, using Chromium and Firefox as open-source programs [30]. In February 2015, Google announced plans to end SPDY support for HTTP / 2. HTTP / 2 was first mentioned when it became clear that SPDY benefited from screen performance (such as Mozilla and nginx) and saw significant improvements in -HTTP / 1. X. After calling the suggestion and interacting with the options, SPDY / 2 was chosen as the reason for HTTP / 2 [11].

Since then, there have been various changes due to discussions in the working group and criticisms of the developers. On February 11, 2016, Google announced that Chrome will no longer support SPDY and NPN after May 15, 2016. RFC7540 reminder. SPDY (pronounced "speedy") is a downloadable open-source network protocol developed by Google for mobile devices. Internet content. SPDY eliminates HTTP traffic with the goal of reducing web page load delays and improving web security. SPDY provides compression delays, replication, and priority reductions, depending on network integration and website delivery conditions. The word "SPDY" is a trademark of Google, not an abbreviation. Throughout the process, major SPDY (figure 1) developers such as Mike Belshe and Roberto Peon were involved in HTTP / 2 development [29].

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

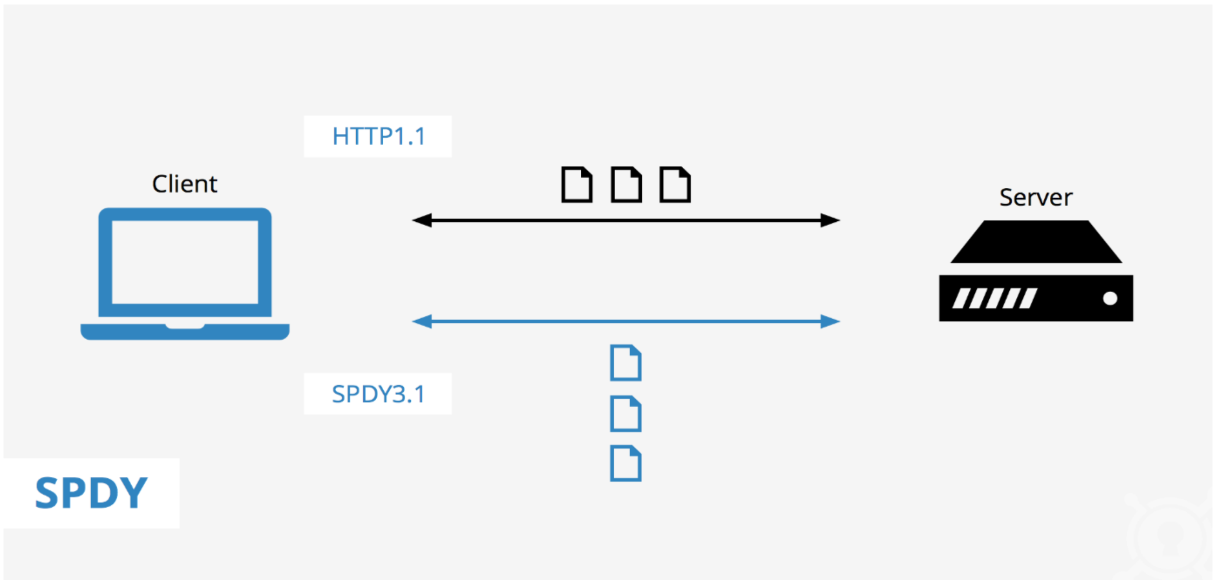


*Fig 1 shows performance of SPDY with respect to HTTP 1.1.*

## 1.3: HTTP/2 (After SPDY)

HTTP / 2 (formerly HTTP / 2.0) is a major update to the HTTP network rules used by the World Wide Web. Quoted from past SPDY test conventions created by Google [9]. HTTP / 2 was created by the Dynamic Hypertext Transfer Protocol for http to (meaning "second") of the Internet Technology Commission. HTTP / 2 is the most important new form of HTTP since HTTP 1.1, which was sent annually in RFC 2068 in 1997. The working group submitted HTTP / 2 to the IESG as proposed in December 2014, and the IESG supported its distribution as proposed. General February 17, 2015. [6] [7] The HTTP / 2 representation was propagated as RFC7540 in May 2015. Stop Effort is supported by Chrome, Opera, Firefox, Internet Explorer 11, Safari, Amazon Silk, and Edge programs. Many major programs added HTTP / 2 help in late 2015 10]. HTTP / 2 makes your application faster, simpler, and more robust by streaming most pre-built HTTP / 1.1 applications within your application and allowing the transport layer itself to address these concerns. (It is a rare combination). It opens entirely new possibilities for improving applications and improving performance. The main goal of HTTP / 2 is to reduce latency by allowing full request replication and response, actively compress HTTP subject fields to reduce log titles, and add significant server support and server push. is. There is considerable support from other protocol developers to meet these requirements. B. New stream control, mistake the executives, and improvement techniques. Nonetheless, these are the main elements that each web engineer requirements to comprehend and apply to their applications [28].

HTTP/2 does not change the semantics of HTTP applications. All logical ideas, for example, HTTP techniques, status codes, URIs, title fields, and so on are generally accessible. All things considered, HTTP/2 indicates how the information is organized and sent between the client and server. Both control the whole cycle and conceal all working framework intricacy inside the new structure. Thus, all current applications can be conveyed inside the application without changing

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

the pre-assembled HTTP/1.1 convention, and these worries are tended to by the vehicle layer itself. Far better, it likewise opens completely additional opportunities for further developing applications and further developing execution. The primary objectives of HTTP/2 are to diminish dormancy by permitting full replication and reaction of solicitations, decrease convention titles by dynamic pressure of HTTP subject fields, and increment application need and server push. Is to add. There is impressive help from other convention designers to meet these necessities. B. New stream control, blunder the executives, and improvement strategies. Notwithstanding, these are the main elements that each web engineer necessities to comprehend and apply to their applications. HTTP/2 does not change the semantics of the HTTP application way. All relevant ideas, for example, HTTP techniques, status codes, URIs, title fields, and so on are dependably accessible. All things considered, HTTP/2 indicates how the information is designed and sent between the client and server. Both control the whole cycle and conceal all working framework intricacy inside the new structure. This implies that all current applications can be conveyed unaltered [18][19].

## 2. Literature Review

### 2.1: HTTP / 1.0- Properties

#### 2.1.1: A comprehensive addressing-scheme

The HTTP protocol uses an orientation point assigned to the Universal Resource Identifier (URI) as a locale (URL) or name (URN), to indicate the system to be used. While making a HTML interface, the URL (Uniform Resource Locator) is a standard structure http://have: port-number/way/file.html. For the most part, the URL reference is a sort of administration://have/file.file-expansion and along these lines, the HTTP convention can utilize fundamental Internet assets. HTTP/1.0 is additionally utilized for correspondence between client specialists and different doors, permitting hypermedia admittance to existing Internet conventions. HTTP/1.0 is intended to permit correspondence with entryways, through facilitating servers, without losing the information communicated by those past arrangements.

#### 2.1.2: Client-Server Architecture

HTTP convention depends on the solicitation/reaction worldview. Correspondence ordinarily happens through a TCP/IP association over the Internet. The default opening is 80, however a few openings can be utilized. This does not forestall the HTTP/1.0 convention from being utilized over some other Internet convention, for however long verification is ensured. The mentioning framework (client) lays out an association with the getting framework (server) and sends the solicitation to the server as a solicitation, URI, and a convention rendition, trailed by a message containing application changes, client subtleties, and conceivable actual substance. The server answers with a status line, which incorporates its own variant of the convention and the achievement or blunder code, trailed by a message containing server data, business subtleties, and conceivable actual substance.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

## **2.1.3: The HTTP / 1.0 Protocol (Connectionless)**

Even though we as of late expressed that a client lays out an association with a server, the convention is known as an association because once a solitary solicitation is fulfilled, the association is downsized. A few conventions generally keep the association open, for example In a FTP meeting you can explore to distant catalogs, and the server monitors what your identity is, and where you are. While this makes the development of the server a lot simpler and liberates you from the functional punishments of the housekeeping meeting, it likewise tracks client conduct, for example ways of exploring between nearby texts, is preposterous. Many web records, if relatively few, incorporate at least one pictures inside the line, and this ought to be downloaded independently, prompting copy associations.

## **2.1.4: The HTTP / 1.0 Protocol (Stateless)**

After the server responds to a client request, communication between the client and the server is disconnected and forgotten. There is no "memory" between communicating with clients. The installation of a pure HTTP server treats every request as if it were new (out of context), that is, it does not store any communication information between actions.

## **2.1.5: MIME Types (Multipurpose Internet Mail Extensions)**

HTTP uses Internet Media Types (formerly MIME Content-Types) to be more open, integrating additional information and typing conversations. In mail applications, where there is no form of communication between the source and the recipient, it is a good idea to draw strong lines in the approved media types. Applications are allowed for the additional opportunity to use unregistered types. At the point when a client sends action to a server, headers are joined that confirm to standard Internet email necessities (RFC 822). Most client applications are hanging tight for a reaction in plain text or HTML. At the point when a HTTP Server communicates data to a client, it presents a header like MIME to tell the client what sort of information is following the header. Interpretation then relies upon the client with the fitting assistance (picture watcher, film player, and so on) related with that kind of information. RFC 2045, displays fields for a variety of topics, including content type. The Content Type field is utilized to characterize the information climate in the MIME business body, by giving the media type and development, and by giving helpful data that might be expected for specific kinds of media. After type and caption, the leftover piece of the header field is essentially a bunch of boundaries, determined in the descriptor/esteem. The request for the boundaries isn't significant.

## **2.1.6: Definition of a Top-Level Media Type**

The depiction of the great quality media type contains:

1. The name and portrayal of a genre varieties, remembering the circumstances for whether a specific group will qualify under that kind,
2. The names and portrayals of the boundaries, if any, are characterized in all sub-sorts (it is required or discretionary to (incorporate whether those boundaries),
3. How the client specialist and/or passage ought to deal with the less popular sorts of this kind.
4. A typical thought for venturesome endeavours of this kind of excellent, if any, and
5. Any limitations on coding for organizations of this kind of top notch.

### 2.1.7: Overview of the Initial Top-Level Media Types

The five discrete top-level media types are:

1. **Text:**

Text data. The "plain" subtype predominantly shows clear text that does not contain arranging orders or guidelines of any sort. Void text is expected to be shown "with no guarantees" and no exceptional programming is expected to be shown by the client specialist. A few more modest renditions will be utilized for rich text in structures where application programming might work on the presence of the text, yet such programming is not expected to find out about the substance. Potential sorts of "text" accordingly, incorporate any word processor design that can be perused without utilizing programming that grasps the arrangement. An extremely basic and convenient subtype, "richtext", was depicted in RFC 1341, with additional surveys on RFC 1896 under "improved".

2. **Image:**

Picture information. "picture" requires a presentation gadget (like a graphical showcase) to see the data. Subtypes are characterized for the most broadly utilized picture organizations, for example, jpeg, gif, png, and so on.

3. **Audio:**

A Sound information. "sound" requires a sound result gadget (like a speaker) to "show" the items. A portion of the various subtypes that are characterized for this sort, are: essential, mp3, wav, and so on.

4. **Video:**

Video information. "video" requires the capacity to show moving pictures, regularly including specific equipment and programming. Subtypes which have a place with this sort's class are: mpeg, mpg, ra, avi, and so on.

5. **Application:**

Addresses different sorts of information, normally non-interpretable paired information or data that will be handled by the application. A little "octet-stream" type is utilized on account of continuous parallel information, where the basic prescribed activity is to compose the subtleties to the client document. The "postscript" and "pdf" subtypes are additionally characterized for the posting of Postscript and PDF material separately. Other expected utilizations of the "application" incorporate calculation sheets, mail-based programming information, and word

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

handling dialects. designs that are not perused straightforwardly. The two main types of media are:

1. **Multipart**

Data covering multiple business types of independent data types. Four subspecies were originally described, which include a sub-standard "mixed" formula that specifies a common set of components, "one" to represent the same data in multiple formats, a "parallel" of components intended for simultaneous viewing, and "grinding" to integrate multiple components. businesses where each component has a default "message / rfc822" type.

2. **Message**:

Secret message. The topic of the media type is the "message" itself or some portion of a specific message type object. Such things could possibly contain different things. The subtype "rfc822" is utilized when the incorporated substance is RFC 822. The subtype "part" is characterized in the RFC 822-part messages, to permit the discontinuity of bodies remembered to be sufficiently enormous to be shipped to a solitary vehicle office. One more modest sort, the "outer body", is characterized by distinguishing bigger subjects by alluding to the outside information source. It ought to be noticed that the kinds of meetings/sub-sorts of meetings led in past sections, are planned to be utilized as a short outline and not as a total manual for the point.

For an inside and out examination of the sorts of web-based media, you are urged to allude to the significant RFCs and internet global positioning frameworks and references. cited from these.

## 2.2: HTTP/1.0 Header Fields

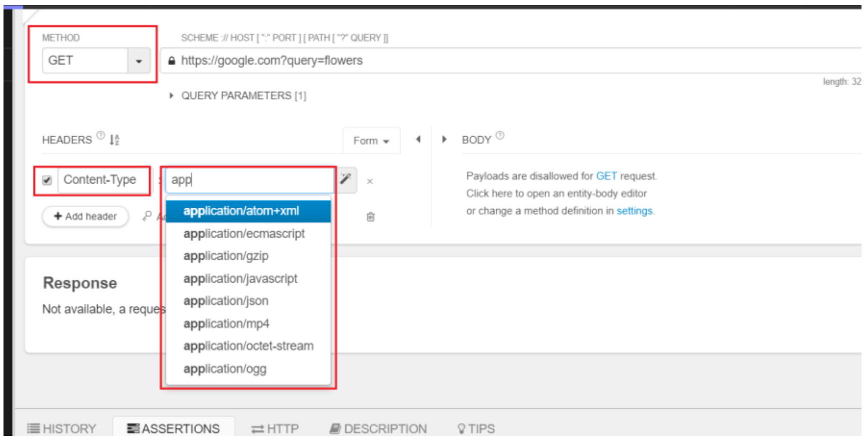
HTTP functionality includes optional headers and explicit information. The title indicates the expected activity on the server, the type of information recovered, the status code, and so on. The rules are even more convenient when you use header fields that are sent over HTTP.

These fields allow you to send unique data from your customers, considering verification, encryption, and / or customer identification. Headers are blocks of information that come before the actual information, and because the data is data, it is often referred to as metadata. If present, the client line gotten from the client is put in the server's CGI variable. - Characters in the title are completely switched over completely to characters. The server might eliminate pre-resolved issues, for example, B. Endorsement, content sort, and content length. Assuming this is significant, the server can decide not to incorporate one or these things on the off chance that the match surpasses the structure's typical cutoff points. This model is HTTP ACCEPT adaptability; another model is the client specialist topic. The kind of MIME client to recover, as determined in the HTTP header. A few guidelines might expect you to recover this information somewhere else. This framework should be all isolated by commas as indicated by HTTP. Topic: Type/Caption, Type/Subtype (eg Confirm: text/xml, text/html). Model: HTTP USER AGEN (or User-Agent header). The program client used to send the solicitation.

Default settings: Programming / Customization / Form. Client Redirect Server: A status code that is displayed regardless of whether the request was successful. Normal error codes indicate that the mentioned document was not found, the prompt was not placed as expected, or confirmation is expected. Actual information. HTTP can send messages to any organization, so it's ready to transfer mixed media such as photo, audio, and video recordings. This perfect way to transfer all kinds of information is one of the most important advantages of HTTP and the Web. It also sends data about the fetch element. Note that the following is not all that means looking at a word, but on the other hand it seems to be okay.

### **2.2.1: Content-Type:**

The Content Type field indicates the type of media information sent to the recipient or, because of the HEAD strategy, the type of media to be sent assuming the request was GET. This field is utilized by programs to decide how they handle information. The client utilizes this data to decide how to deal with a video record or in-line picture. There are various sorts of content kind in figure 2:



METHOD: GET
SCHEME :// HOST [ ":" PORT ] [ PATH [ "?" QUERY ]]: https://google.com?query=flowers
length: 32
QUERY PARAMETERS [1]
HEADERS: Content-Type: app
application/atom+xml
application/ecmascript
application/gzip
application/javascript
application/json
application/mp4
application/octet-stream
application/ogg
BODY: Payloads are disallowed for GET request. Click here to open an entity-body editor or change a method definition in settings.
Response
Not available, a reques

*Fig 2 shows different types of Content-Type available for HTTP requests.*

### **2.3: HTTP/1.0 Methods**

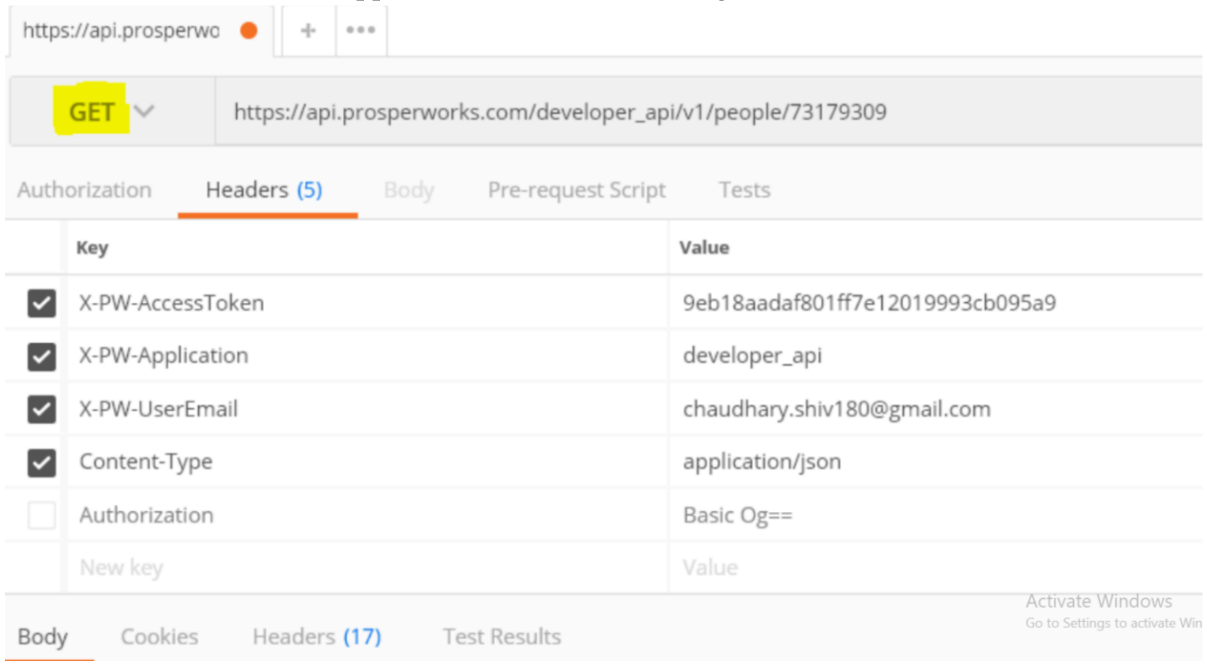
HTTP / 1.0 provides an unconventional set of strategies that should be used to show the reason for the request. The three most used strategies are GET, HEAD, and POST.

### **2.3.1: The GET Method**

The GET strategy (figure 3) is used to search for a specific report - when you click on a link, it uses GET. GET ought to most likely be utilized when URL access will not change the situation with the site (for instance, by adding or erasing data) and that SEND ought to be utilized. The semantics of the GET method shifts to "Contingent GET" where the message includes the title "When-Fixed-From:". The restrictive GET strategy demands that the predetermined assistance be moved if it is transformed from the date given the title "If-Modified-Since:". A limited GET

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

strategy is designed to reduce network usage by allowing retained organizations to be updated without the need for various applications or to submit insignificant information.


https://api.prosperwo

GET https://api.prosperworks.com/developer_api/v1/people/73179309

Authorization | Headers (5) | Body | Pre-request Script | Tests

| | Key | Value |
|---|---|---|
| ✓ | X-PW-AccessToken | 9eb18aadaf801ff7e12019993cb095a9 |
| ✓ | X-PW-Application | developer_api |
| ✓ | X-PW-UserEmail | chaudhary.shiv180@gmail.com |
| ✓ | Content-Type | application/json |
| | Authorization | Basic Og== |
| | New key | Value |

Body | Cookies | Headers (17) | Test Results

*Fig 3 shows a simple example of GET request.*

## **2.3.2: The HEAD Method**

The HEAD method is used to query data about just a report, not an actual record. HEAD is a lot quicker than GET, as a tiny measure of information is sent. It's not unexpected utilized by clients utilizing store, to check whether the report has changed as it was last gotten to. On the off chance that not, the neighbourhood duplicate can be reused, generally the refreshed adaptation should be returned through GET. The meta-information contained in headers considering a HEAD solicitation ought to be equivalent to the data sent because of a GET demand. This technique can be utilized to get extra data about the application distinguished URI of the application without moving the actual information. This technique is many times used to check hypertext joins for official, open, and ongoing changes.

## **2.3.3: The POST method**

Same way to perform tasks such as B.: A note about available resources. Post a message to a bulletin board, newsgroup, mailing list, or similar newsgroup. Provide a block of data (usually a form) in the data management process. Expand your website with additional features. This is a "SEND" query to the live filesystem of "/ ~ hy556 / cgi-bin / post-query", which simply reflects the value received. The client writes a list of MIME types that it can build to receive and suggests the adaptation of the WWW library it is utilizing. At last, it shows the MIME type used to record the information sent, the quantity of characters entered, and a rundown of factors and their qualities gathered from the client. The MIME rendition of "utilization/x-www-structure URL encoded" implies that different worth matches are encoded utilizing the URL encoding technique. Every extraordinary person, including accentuation, are encoded. nn, where nn is the ASCII worth of the hexadecimal person. The server consents to utilize HTTP 1.0 interpretation to engender and send 200 situations with that it has effectively handled the client demand. By then, send the date and detach as an Apache HTTP server. It also displays the data to be sent using the MIME 1.0 customization and also shows that it contains MIME variations of the data that is exported without "Content Type:". Finally, send the number of characters to send, followed by a clear line and the actual information. The client and server headers are RFC822 compliant email subject lines. The client can send the title (comma or other) "confirmation:" and the server must convert the information to the client type.

## **2.4: HTTP/1.1 (The Next Generation)**

The simplicity of HTTP was a key factor in its rapid adoption, but its simplicity was a major drawback. HTTP / 1.0 does not properly consider tiered proxies, temporary storage, the need for continuous connectivity, or the impact of virtual strangers. In addition, the development of a fully loaded application called "HTTP / 1.0" required changing the protocol version so that the two communication applications determine each other's actual capabilities. HTTP / 1.1 (RFC 2616) replaces HTTP / 1.0 with very high performance.

### **2.4.1: Determined Connection's**

A significant change between HTTP/1.1 and past HTTP variants is that consistent correspondence is the default conduct for HTTP associations. That is, except if generally indicated, the client should accept that the server keeps a continuous association even after a mistake reaction from the server. Diligent associations give a way to clients and servers to flag the end of TCP associations. This mark happens utilizing the Connection1 field. When the closure is marked, the client should not make any further demands on this association. Preceding the constant association, a different TCP association was laid out to download every URL.

This increases the load on the HTTP server and causes Internet congestion. To use online images and other related data, clients often need to make multiple requests to the same server in a short amount of time. There are many advantages to continuous HTTP connections.

Opening and closing multiple TCP organizations also saves CPU time on the switch (client, server, connector, access channel, corridor, or archive) and saves the memory used by TCP conference control Blocks in the system editor. HTTP requests and responses can be sent through

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

your organization. The pipe feature allows clients to make different requests without waiting for each response, making TCP connections more productive and significantly reducing idle time. Network deadlocks are reduced by reducing the number of packets caused by TCP openness (three-way data transmission) and allowing enough TCP time to determine the status of network deadlocks. No time is spent opening the handshake in the TCP organization, reducing the delay of the next request. HTTP can advance well indeed, as mistakes can be accounted for without a closure punishment TCP association. Clients utilizing future forms of HTTP may ideally attempt another component, yet while associating with a more seasoned server, attempt once more with the old semantics after a mistake was accounted for. To set the essential association field to close, persistent correspondence is switched off in client and server correspondence by means of HTTP/1.1 (subsequently, HTTP/1.0 way of behaving).

The HTTP/1.1protocol covers various things expected to make impermanent capacity conceivable. Saving time would be futile on the off chance that it did not fundamentally further develop execution. The objective of brief stockpiling on HTTP/1.1 is to kill the need to send demands as a rule, and to take out the need to send full reactions much of the time. That is, there are two primary justifications for why web storing is utilized:

Reduction is postponed because it keeps the need (closer to the client) than the original server, it requires a small investment for the client to track down the article and show it. This makes the sites seem overly responsive.

## <u>**2.5: SPDY: After HTTP**</u>

The SPDY project defines and uses a web application layer protocol that greatly diminishes latency. The highest SPDY standards are:

• Identify 50% reduction in page load time. Our first results are closer to this goal (see below).

• Reduce the difficulty of shipping. SPDY uses TCP as a basic transport platform, so it does not require variations to existing communications infrastructure.

• Stay away from the requirement for any progressions to the substance of the site creators. Major changes expected to help SPDY (figure 4) are in the client and web server applications.

• Bringing together the same people who are willing to look at arrangements to deal with the problem of procrastination. We wish to promote this new conference in partnership with local experts and industry experts.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

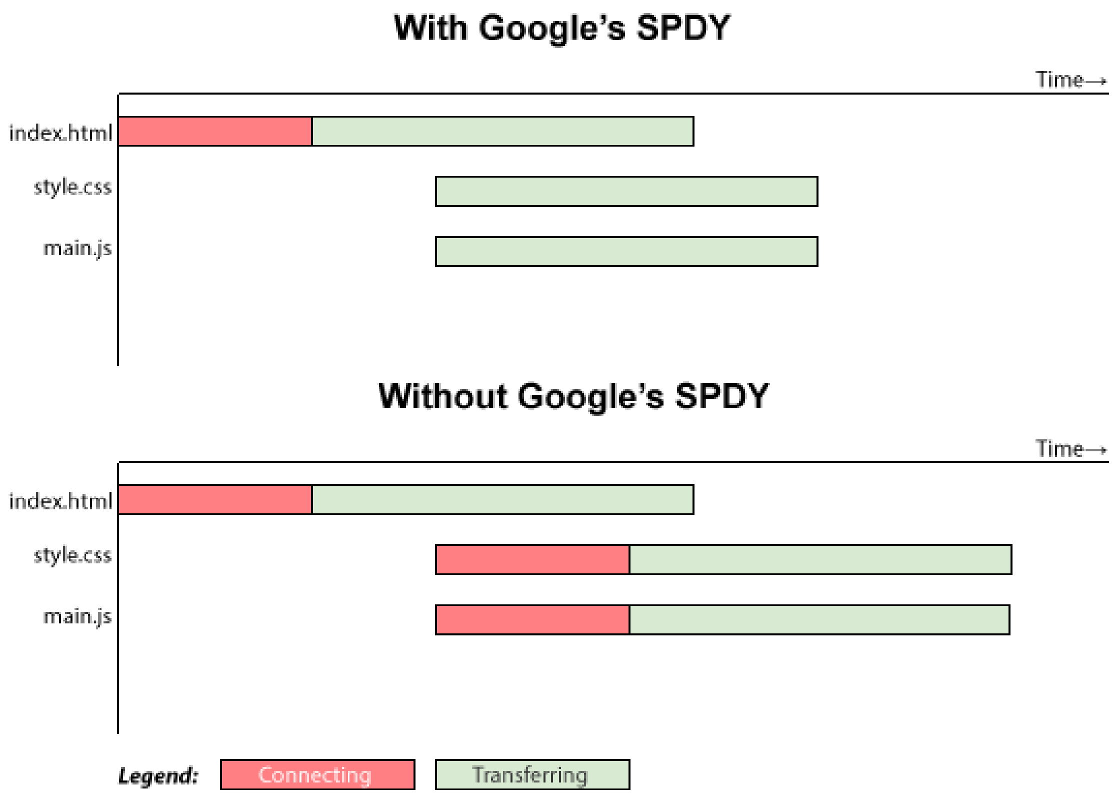


*Fig 4 shows the comparable of the speed of SPDY.*

Some specific technical objectives are:

• Allowing multiple HTTP requests at once to run in one TCP session.

• Reduce the bandwidth currently used by HTTP by pressing titles and removing unnecessary topics.

• Define an easy-to-use and efficient server protocol. We hope to reduce the complexity of HTTP by limiting border controls and defining easy-to-send message formats.

• Making SSL a basic transport protocol, for better security and compatibility with existing network infrastructure. Although SSL introduces delay fees, we believe that the long-term future of the web depends on secure network connections. Additionally, the use of SSL is required to ensure that the connection to all existing proxies is broken.

• Enables the server to initiate client communication and push client data whenever possible.

• Bandwidth efficiency is low. Although the efficiency of dialling bandwidth is close to 90%, the efficiency of the connection speed is about ~ 32%.

• SSL poses certain challenges for delays and transfers. Among these are: an additional SSL shake for RTT; crucifixion; an attempt to cache some proxies. We want to do more SSL tuning.

• Our unfortunate bulk results have not been verified. Albeit a ton of parcel misfortune research has been finished, we need more information to fabricate a virtual bundle misfortune model on the web. We want to gather this information to give more exact bundle misfortune reproduction.

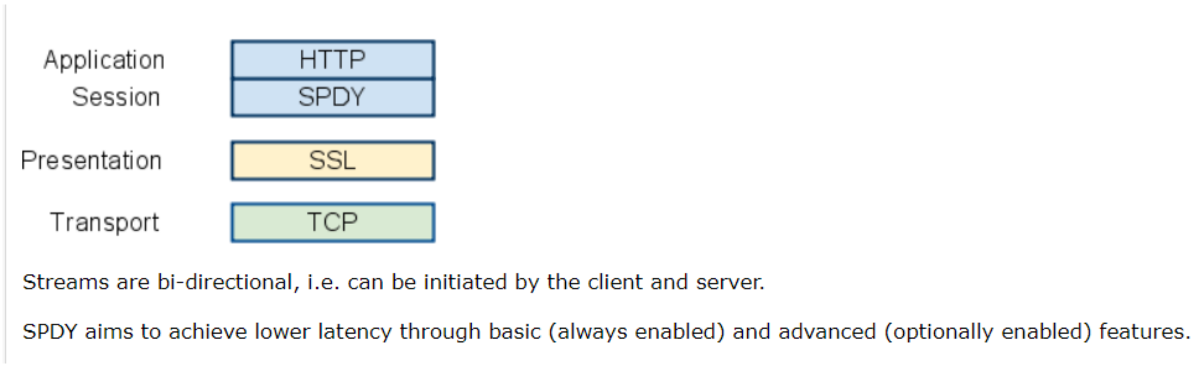
TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

• Distinguishing a solitary SPDY association misfortune now and then doesn't function admirably on numerous associations. That is, opening various associations is still quicker than losing a solitary association when RTT is extremely high. We want to find out when it is suitable for the SPDY client to make another association or close an old association and what this could have to do with the SERVER.

• The server can use more ingenuity than we have built so far. We need more research on server-initiated broadcast areas, client network information to pre-download suggestions, and more.

## **2.5.1: SPDY design and structures**

SPDY adds a high-density SSL layer that considers multiple transfers simultaneously, broken between one TCP organization.
Standard HTTP GET and POST formats remain unchanged (figure 5); however, SPDY specifies a new independent encryption format and telephone data.



Streams are bi-directional, i.e. can be initiated by the client and server.

SPDY aims to achieve lower latency through basic (always enabled) and advanced (optionally enabled) features.

*Fig 5 shows SPDY implementation in TCP layers.*

## **2.5.2: Basic structures:**

- **Multiplexed streams**

SPDY permits limitless well-matched streaming over a single TCP connection. Because packages are disconnected from one channel, the performance of TCP may be very high: low community connection wishes to be made, and few, however very dense, packets are to be had issued.

- **Request ordering**

Although unlimited parallel streams solve a string problem, it introduces another: if channel bandwidth is blocked, the client may block requests for fear of closing the channel. To resolve this issue, SPDY uses the application theme: the client can request as many items as they want from the server, and provide the key for each application. This prevents the network channel from being overloaded with unnecessary resources while waiting for the most important request.

- **HTTP header compression**

SPDY wrap application and response to HTTP headers, resulting in fewer packets.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

## 1. Advanced structures

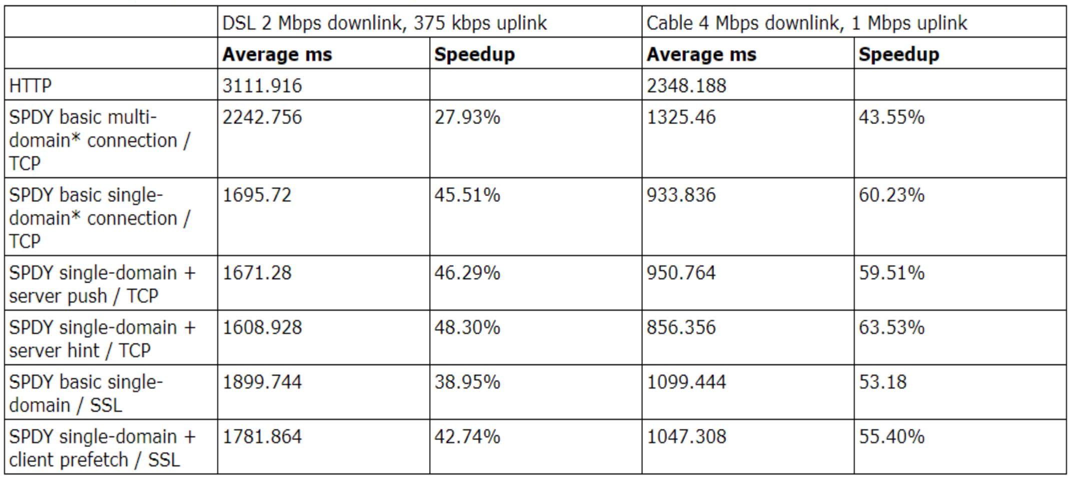
Moreover, SPDY gives a high-level element, server-started streaming. The embedded live stream of the server can be used to transfer content to the client without the client saying so. This option is designed for the web developer in two ways:

- **Server push**

SPDY testing for selecting servers that deliver information to customers using the X-Linked Content header. This article informs the client that the server is pushing the client before the client requests it. By downloading the forecast page (for example at the time a client first visits a site), this can further improve client information.

- **Server hint**

As opposed to naturally pushing client assets, the server utilizes X-Sub assets to recommend that the client ought to demand specific administrations, in situations where the server realizes ahead of time that the client will be required. In any case, the server will in any case hang tight for the client demand prior to sending the substance. With more slow connections, this choice can decrease the time it takes for the client to figure out (figure 6) they need the assistance by many milliseconds, and can be better at non-original page loading.



| | DSL 2 Mbps downlink, 375 kbps uplink | | Cable 4 Mbps downlink, 1 Mbps uplink | |
|---|---|---|---|---|
| | **Average ms** | **Speedup** | **Average ms** | **Speedup** |
| HTTP | 3111.916 | | 2348.188 | |
| SPDY basic multi-domain* connection / TCP | 2242.756 | 27.93% | 1325.46 | 43.55% |
| SPDY basic single-domain* connection / TCP | 1695.72 | 45.51% | 933.836 | 60.23% |
| SPDY single-domain + server push / TCP | 1671.28 | 46.29% | 950.764 | 59.51% |
| SPDY single-domain + server hint / TCP | 1608.928 | 48.30% | 856.356 | 63.53% |
| SPDY basic single-domain / SSL | 1899.744 | 38.95% | 1099.444 | 53.18 |
| SPDY single-domain + client prefetch / SSL | 1781.864 | 42.74% | 1047.308 | 55.40% |

*Fig 6 shows average page load times for different websites.*

There has been a growing emphasis on website uploading when it comes to user information and its potential impact on Google search engine rankings. This is because Google is increasingly critical of the website loading campaign. Remember that in Google Webmaster Tools, a new category has just been added. Reporting site performance on page loading time. Not only that, but Google has a "Speed Page" (http://code.google.com/speed/page-speed/), a Firefox plug-in that will measure Web page load speeds. Speed Score, "Scale of 1-100. If you are serious about making your websites easier for both users and Google, you should look for ways to improve your page loading time. Slow websites bring bad user information, and websites provide bad information to top Google visitors.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

## 2.6: HTTP/2: After HTTP1 and SPDY

In order to achieve the performance standards, set by the HTTP Working Group, HTTP / 2 introduces a new non-reversible binary platform with previous HTTP / 1.x servers and clients — which is why a large version of the protocol is growing in HTTP / 2.

Unless you are using a web server (or a custom client) that runs with green TCP sockets, then you will not see the difference: every new, substandard framework is created by the client and server on your behalf. The only noticeable difference will be improved performance and acquisition of new skills such as prioritizing the application, flow control, and server push.

A brief history of SPDY and HTTP / 2

SPDY was a test protocol, developed by Google and announced in mid-2009, its main objective being to try to reduce web page load delays by addressing other known HTTP / 1.1 performance limitations. Specifically, the project objectives identified are as follows:

• Identify a 50% lessening in page load time (PLT).
• Avoid the need for any changes in the content of the website authors.
• Reduce the complexity of the application, and avoid changes in network infrastructure.
• Develop this new process in partnership with the open-source community.
• Collect authentic performance data to (in) authenticate the test protocol.

## 3. Methodology

Albeit a few fundamental standards have been refreshed throughout the long term (FTP became SFTP; POP3 switched over completely to IMAP; and telnet became SSH), HTTP/1.1 has not changed and worked on numerous issues with speed, security, and ease of use subsequently. In 2015, be that as it may, the Internet Engineering Task Force (IETF) carried HTTP/2, the second biggest rendition of the most helpful Internet convention, HTTP. Here is a glance at why it is great, why it is so great, and a few different ways you can begin utilizing it today.

1.1. SPDY as a pioneer to HTTP/2

Google was quick to explore issues with HTTP/1.1. At that point, they were burning through great many dollars a year to help their server farms, and the HTTP/1.1 convention is essentially too costly concerning CPU assets and web network. They have redesigned SPDY as another HTTP/1.1 test strategy a convention intended for better security and further developed page stacking times that might go before HTTP/2.

### 3.1 HTTP/2 PROS

2. What are nearly the benefits of using HTTP / 2?
3. • HTTP/2 is BINARY, rather than text.
4. • HTTP / 2 is completely duplicated. This implies that HTTP/2 can send different information demands related to a solitary TCP association. This is the most exceptional element of the

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

HTTP/2 convention since it permits you to download web documents in A Sync mode from a solitary server. Most present-day programs limit TCP network to a solitary server.

5. • It usages HPACK header pressure to bring it down.
6. • Permits servers to "push" replies continuously into a client repository rather than waiting for a new application for each application.
7. • Usages a new ALPN extension that takes into account quicker scrambled associations as the still up in the air during the underlying association.
8. • Reduce overtime (RTT), allowing your website to load faster without fully performing.
9. • Domain segregation and asset mergers are no longer required via HTTP / 2.

### 3.2 HTTP/2 CONS

Many professionals around the world were expecting a lot of new things from the HTTP / 2 conference, however these were not included in the final form. The principal reason is straightforward: HTTP/2 ought to keep up with in reverse similarity with more seasoned HTTP/1.1, utilizing a similar POST and GET applications, status codes (200, 301, 404, and 500), and so on. Likewise, a couple of new highlights. as clog page headers are in danger of BREACH and CRIME assaults.
To start with, there could be no alternate way accessible today that is higher than HTTP/2. Programs would not uphold SPDY for HTTP/2, while just Mozilla Firefox upheld SPDY. Notwithstanding, as an IT proficient, you ought to in any case know about the shortcomings of the convention. A few specialists accept that these issues could be settled in the future with the arrival of the "HTTP/3" convention, however for the present, these are only a couple of the risks. It is not very fast and it is not modern.

**Encryption is not required.** In recent years, data encryption has been severely limited in monetization. But hacks and government surveillance are effective threats, which affect your personal and business life.

Encryption may protect you from those types of threats, but conflicts within the HTTP / 2 developer team have led to the decision to "leave encryption as it is in HTTP / 1.1.". This means that website owners can still opt for a lower level of security, which may put users at risk. Browser developers can “fix” this problem correctly - for now, they only turn on HTTP / 2 if SSL / TLS protocols are available.

**Cookie security is motionless an issue.** Encryption may protect you from those types of threats, but conflicts within the HTTP / 2 developer team have led to the decision to "leave encryption as it is in HTTP / 1.1.". This means that website owners can still opt for a lower level of security, which may put users at risk. Browser developers can "fix" this problem correctly - for now, they only turn on HTTP / 2 when SSL / TLS protocols are available.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

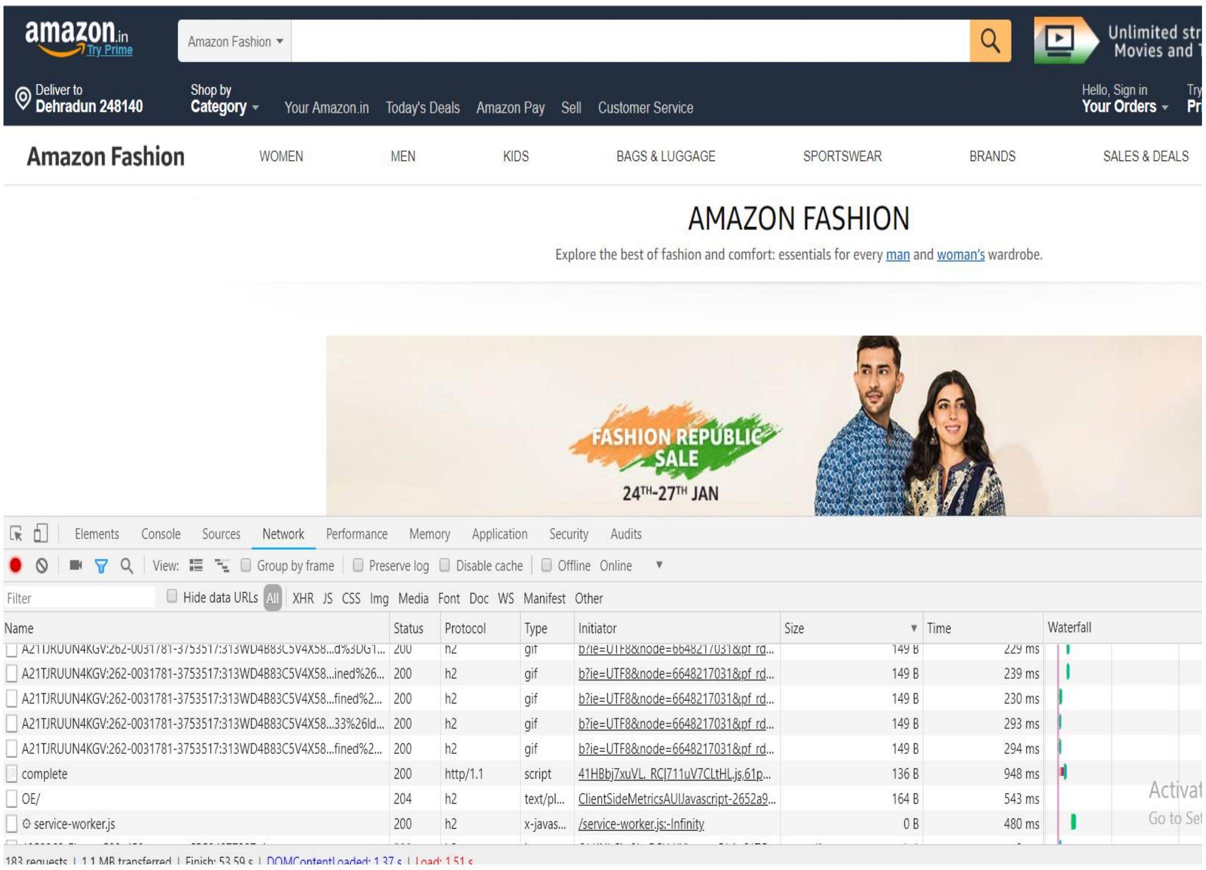
This may not sound awful, yet the truth of the matter is that treats might be taken or hacked by crooks. This implies programmers can get to your email, virtual entertainment, and different sites that contain your own information, even without passwords. This is called cross-site prearranging (XSS). Specialists have trusted that treats will be supplanted with new innovations, however up to this point they have not.

## 4. Results

Impact of HTTP / 2 performance on standard websites, and while using HTTP / 2 improves page load speed and efficiency, increasing cookie usage makes it less secure and, in this context, I checked out top websites from Alexa and found differences between speed, status, time, and file size using Google Chrome Developer Tools and Console.

**4.1 Speed of HTTP /2:** The main purpose of this analysis is to evaluate the impact of the new HTTP/2 standards on the web. Carefully orchestrated, real-time web-based tests deliver the fastest pages on HTTP/2 with legally permitted copying and compression in simulated, real-world online scenarios. To increase transport layer encryption was intentionally disabled in stringent HTTP/1.1 benchmark tests. In figure 7 we can see that the result of speed of HTTP/2.

The first website we used was www.amazon.in and found the following result:



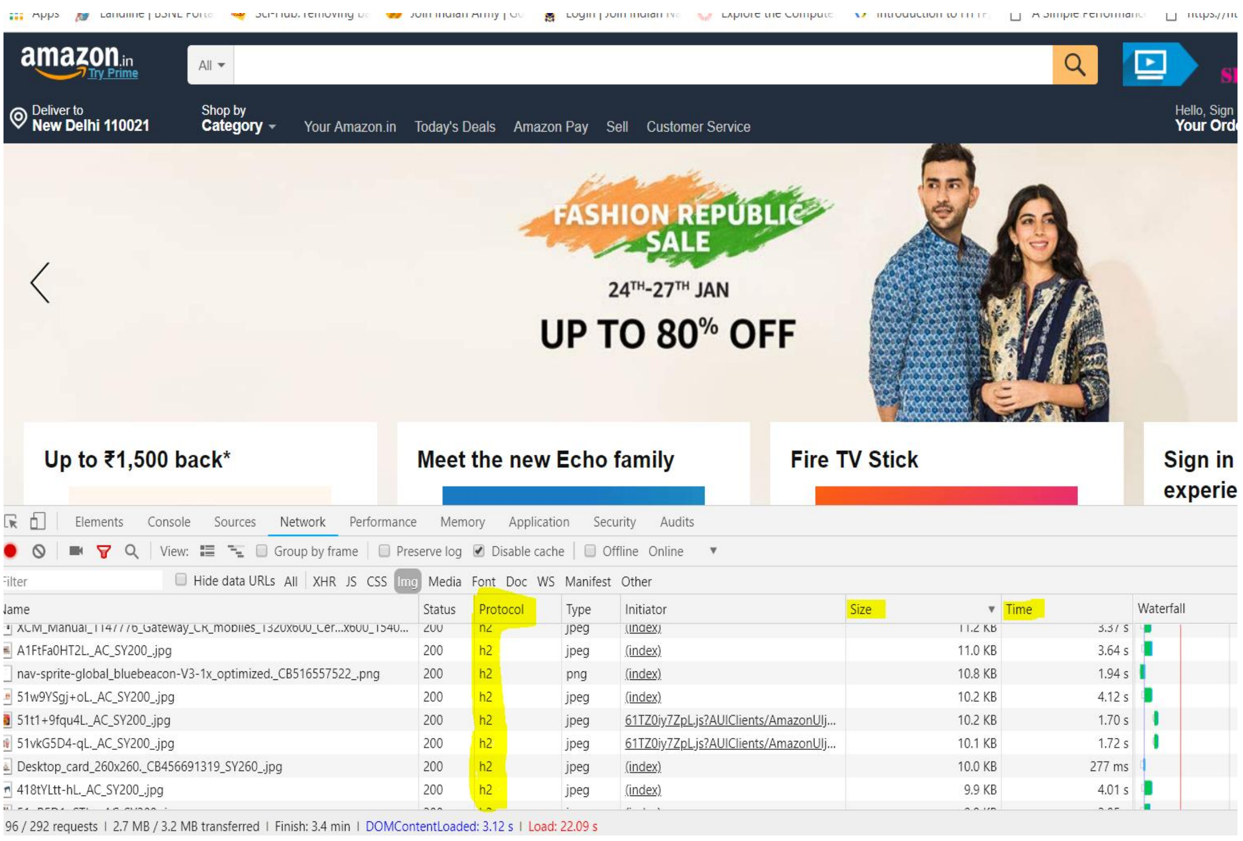
| Name | Status | Protocol | Type | Initiator | Size | Time |
|---|---|---|---|---|---|---|
| A21TJRUUN4KGV:262-0031781-3753517:313WD4B83C5V4X58...d%3DG1... | 200 | h2 | gif | b?ie=UTF8&node=6648217031&pf_rd... | 149 B | 229 ms |
| A21TJRUUN4KGV:262-0031781-3753517:313WD4B83C5V4X58...ined%26... | 200 | h2 | gif | b?ie=UTF8&node=6648217031&pf_rd... | 149 B | 239 ms |
| A21TJRUUN4KGV:262-0031781-3753517:313WD4B83C5V4X58...fined%2... | 200 | h2 | gif | b?ie=UTF8&node=6648217031&pf_rd... | 149 B | 230 ms |
| A21TJRUUN4KGV:262-0031781-3753517:313WD4B83C5V4X58...33%26ld... | 200 | h2 | gif | b?ie=UTF8&node=6648217031&pf_rd... | 149 B | 293 ms |
| A21TJRUUN4KGV:262-0031781-3753517:313WD4B83C5V4X58...fined%2... | 200 | h2 | gif | b?ie=UTF8&node=6648217031&pf_rd... | 149 B | 294 ms |
| complete | 200 | http/1.1 | script | 41HBbj7xuVL._RC|711uV7CLtHL.js,61p... | 136 B | 948 ms |
| OE/ | 204 | h2 | text/pl... | ClientSideMetricsAUIJavascript-2652a9... | 164 B | 543 ms |
| service-worker.js | 200 | h2 | x-javas... | /service-worker.js:-Infinity | 0 B | 480 ms |

183 requests | 1.1 MB transferred | Finish: 53.59 s | DOMContentLoaded: 1.37 s | Load: 1.51 s

*Fig 7 shows the speed of http/1.1 and http2 and the speed differences between them.*

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

**4.2 Impact of the HTTP/2 in Packet Loss:** You can use different sites on the network to share hotspots that negatively impact HTTP/2 performance and use the results of other available SPDY tests. HTTP/2 is vulnerable to packet loss. Overcrowding in TCP networks has been shown to be prone to packet loss, disrupting the entire HTTP/2 flow. In this case, it is recommended to use migration to process the packets to avoid persistent output. In figure 8 shows the HTTP/2 size and speed ratio. The Waterfall shows the collapse of the data.



| Name | Status | Protocol | Type | Initiator | Size | Time | Waterfall |
|---|---|---|---|---|---|---|---|
| XCM_Manual_1147776_Gateway_CR_mobiles_1320x600_Cer...x600_1540... | 200 | h2 | jpeg | (index) | 11.2 KB | 3.37 s | |
| A1FtFa0HT2L._AC_SY200_.jpg | 200 | h2 | jpeg | (index) | 11.0 KB | 3.64 s | |
| nav-sprite-global_bluebeacon-V3-1x_optimized._CB516557522_.png | 200 | h2 | png | (index) | 10.8 KB | 1.94 s | |
| 51w9YSgj+oL._AC_SY200_.jpg | 200 | h2 | jpeg | (index) | 10.2 KB | 4.12 s | |
| 51t1+9fqu4L._AC_SY200_.jpg | 200 | h2 | jpeg | 61TZ0iy7ZpL.js?AUIClients/AmazonUIj... | 10.2 KB | 1.70 s | |
| 51vkG5D4-qL._AC_SY200_.jpg | 200 | h2 | jpeg | 61TZ0iy7ZpL.js?AUIClients/AmazonUIj... | 10.1 KB | 1.72 s | |
| Desktop_card_260x260._CB456691319_SY260_.jpg | 200 | h2 | jpeg | (index) | 10.0 KB | 277 ms | |
| 418tYLtt-hL._AC_SY200_.jpg | 200 | h2 | jpeg | (index) | 9.9 KB | 4.01 s | |

96 / 292 requests | 2.7 MB / 3.2 MB transferred | Finish: 3.4 min | DOMContentLoaded: 3.12 s | Load: 22.09 s

*Fig 8 shows the HTTP/2 size and speed ratio. The Waterfall shows the collapse of the data.*

**4.3 Impact of the HTTP/2 in time:** We can also view server dispatch and priority based on current HTTP/2 usage. figure 9 below shows the average page load time of various

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**

global websites for different data content such as HTML, CSS, JS etc.

| | Website | HTML | CSS | JS | Images |
|---|---|---|---|---|---|
| 1 | Google | 2 | 1 | 5 | 4 |
| 2 | Youtube | 7 | 6 | 18 | 45 |
| 3 | Wikipedia | 3 | 2 | 5 | 5 |
| 4 | Yahoo | 4 | 2 | 8 | 38 |
| 5 | Baidu | 9 | 0 | 13 | 16 |
| 6 | Amazon | 4 | 28 | 40 | 45 |
| 7 | Taobao | 7 | 11 | 55 | 93 |
| 8 | QQ | 21 | 1 | 29 | 132 |
| 9 | Weibo | 7 | 4 | 11 | 20 |
| 10 | Tmall | 8 | 6 | 8 | 69 |
| 11 | Ebay | 9 | 5 | 8 | 76 |
| 12 | Hao123 | 7 | 2 | 20 | 57 |
| 13 | Yandex | 2 | 0 | 6 | 12 |
| 14 | Sohu | 73 | 4 | 93 | 219 |
| 15 | Bing | 4 | 0 | 13 | 6 |

*Fig 9 demonstrations different page load time data of dissimilar websites.*

**4.3.1 Impact of the HTTP/2 difference page load time:** The difference between the page load time on these dissimilar websites. In figure 10 we show difference between page load times of different websites:

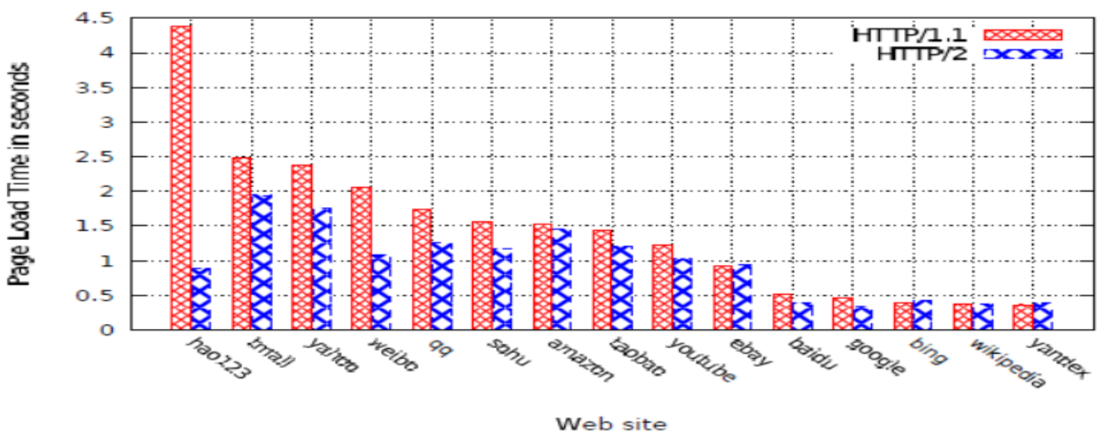
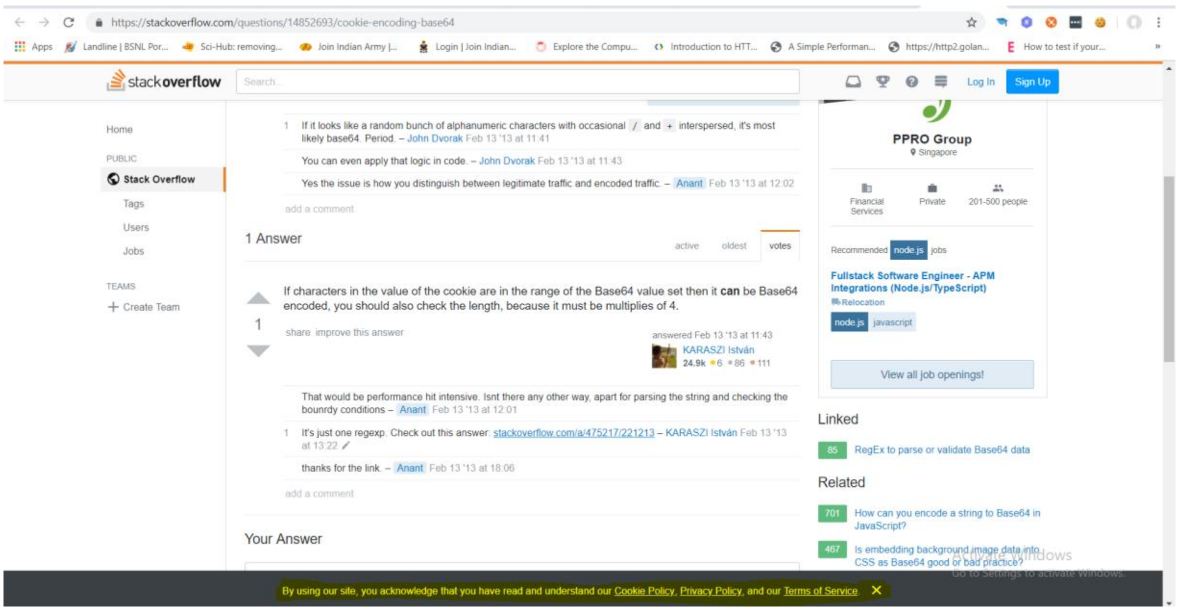


*Fig 10 shows difference between page load times of different websites.*

**4.4 Impact of the HTTP/2 store cookies:**

I took the results from various top websites that switched to HTTP / 2 and allowed to store cookies in the local web browser history especially the repository (figure 11). Below are screenshots of various websites:

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.04, I.03 (2025)
**https://www.tejasjournals.com/**
**https://doi.org/10.63920/tjths.43002**



1 If it looks like a random bunch of alphanumeric characters with occasional / and + interspersed, it's most likely base64. Period. – John Dvorak Feb 13 '13 at 11:41

You can even apply that logic in code. – John Dvorak Feb 13 '13 at 11:43

Yes the issue is how you distinguish between legitimate traffic and encoded traffic. – Anant Feb 13 '13 at 12:02

add a comment

1 Answer

1 If characters in the value of the cookie are in the range of the Base64 value set then it **can** be Base64 encoded, you should also check the length, because it must be multiplies of 4.

answered Feb 13 '13 at 11:43
KARASZI István
24.9k ●6 ●86 ●111

That would be performance hit intensive. Isnt there any other way, apart for parsing the string and checking the bounrdy conditions – Anant Feb 13 '13 at 12:01

1 It's just one regexp. Check out this answer: stackoverflow.com/a/475217/221213 – KARASZI István Feb 13 '13 at 13:22

thanks for the link. – Anant Feb 13 '13 at 18:06

add a comment

Your Answer

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service. ✕

*Fig 11 shows website stack overflow uses cookies.*

## 5. Conclusions

The core purpose of this analysis was to assess the impact of the new HTTP / 2 structures on the Web. Well-planned, real-time, web-based testing provides the fastest page load over HTTP / 2, with legally permitted replication and compression under simulated, realistic online scenarios. increase. At the transport layer, encryption was intentionally disabled during a strict HTTP / 1.1 comparison test. You can use diverse network settings to categorize the key parameters that negatively impact HTTP / 2 performance and validate the results of other present SPDY tests. HTTP / 2 carries the risk of packet loss. Overcrowding in TCP networks is clearly associated with packet loss, affecting all over HTTP / 2 streams. This situation proposes using the transport layer protocol for loss of packet to maintain stable output.

We were also able to show server pushes and prioritization based on current HTTP / 2 usage. The results show that these approaches can lead to significant improvements in repetition. Though, this is still a relatively small area of HTTP / 2. This is primarily because the definition gives developers a lot of freedom.

Also, these algorithm's require diverse configurations and tweaks to each website. There is an urgent need to protect cookies and make them more secure with browser-level encryption. Misuse of cookies in your browser can be dangerous because very useful information such as usernames and passwords are stored in cookies. Time cookies also store your browsing history and allow you to reactivate the time when a user logs in to a password-protected website such as a bank or email payment website.

## References:

[1]. H. De Saxce, I. Oprescu, and Y. Chen, "Is HTTP/2 really faster than HTTP/1. 1?" *Proc. - IEEE INFOCOM*, vol. 2015–Augus, pp. 293–299, 2015.

[2]. A. Banerjee, L. K. Chong, S. Chattopadhyay, and A. Roychoudhury, "Detecting energy bugs and hotspots in mobile apps," *Proc. 22nd ACM SIGSOFT Int. Symp. Found. Software. Eng. - FSE 2014*, vol. 3, pp. 588–598, 2014.

[3]. T. Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update , 2010 – 2015," *Growth Lakel.*, vol. 2011, no. 4, pp. 2010–2015, 2011, https://www.ieee802.org/3/ad_hoc/bwa/public/sep11/nowell_01_0911.pdf.

[4]. H. Brotherton, *Data Center Energy Efficiency Calculator.* PhD thesis, Purdue University, 2014.

[5]. F. Yang, P. Amer, J. Leighton, and M. Belshe, "A Methodology to Derive SPDY ' s Initial Dictionary for Zlib Compression," 2012.

[6]. Mike Belshe and Roberto Peon, "SPDY Protocol - Draft 1 - The Chromium Projects," 2012. [Online]. Available: https://www.chromium.org/spdy/spdy-protocol/spdy-protocol-draft1.

[7]. David., "Difference between HTTP 1.0 and 1.1 | Difference Between.Net," 2013. [Online]. Available: http://www.differencebetween.net/technology/protocols-formats/difference-between-http-1-0-and-1-1/.

[8]. A. S. Alshammari and A. Al-Mogren, "HTTP/2 in Modern Web and Mobile Sensing-based Applications Analysis, Benchmarks and Current Issues," *J. Electr. Electron. ...*, vol. 5, no. 3, 2016, https://www.hilarispublisher.com/open-access/http2-in-modern-web-and-mobile-sensingbased-applications-analysisbenchmarks-and-current-issues-2332-0796-1000193.pdf.

[9]. Google., "Make the Web Faster | Google Developers." [Online]. Available: https://developers.google.com/speed/?csw=1.

[10]. Chrome, "A 2x Faster Web. [online] chromium blog. available at: http://blog.chromium.org/2009/11/2x-faster-web.html," 2009.

[11]. M. P. Matt Welsh, Ben Greenstein, "SPDY Performance on Mobile Networks | PageSpeed Insights Google Developers," 2012. [Online]. Available: https://developers.google.com/speed/articles/spdy-for-mobile.

[12]. D. Stenberg, "HTTP2 Explained," *Acm Sigcomm Comput. Commun. Rev.*, vol. 44, no. 3, pp. 120–128, 2014, https://dl.acm.org/doi/10.1145/2656877.2656896.

[13]. "Usage Statistics of SPDY for Websites, Feb 2017." [Online]. Available: https://w3techs.com/technologies/details/ce-spdy/all/al

[14]. 3GPP TS23.501, "System Architecture for the 5G System", v.15.2.0, Jun.2018.

[15]. 3GPP TR 29.890, "Study on CT WG3 Aspects of 5G System Phase 1", v.15.1.0, March 2018.

[16]. The Road to 5G: Drivers, Applications, Requirements and Technical Development, GSA, Washington, DC, USA, Nov. 2015.

[17]. Ayush Kashyap et al., Design and Implementation of an Intelligent Loan

Eligibility System Using Machine Learning Techniques, TEJAS Journal of Technologies and Humanitarian Science, ISSN-2583-5599, Vol.04, I.02 (2025), https://doi.org/10.63920/tjths.42002

[18]. Nokia, "Security challenges and opportunities for 5G mobile networks," 2017.

[19]. C. X. Wang et al., "Cellular architecture and key technologies for 5G wireless communication networks," IEEE Communications Magazine, vol. 52, no. 2, pp. 122-130, 2014.

[20]. P. K. Agyapong, M. Iwamura, D. Staehle, and W. Kiess, "Design considerations for a 5G network architecture," IEEE Communications Magazine, vol. 52, no. 11, pp. 65-75, 2014.

[21]. Gupta and R. K. Jha, "A Survey of 5G Network: Architecture and Emerging Technologies," IEEE Access, vol. 3, pp. 1206-1232, 2015.

[22]. R. Modarressi and R. A. Skoog, "Signaling System No.7: a tutorial," IEEE Communications Magazine, vol. 28, no. 7, pp. 19-20, 1990.

[23]. IETF RFC7540, “Hypertext Transfer Protocol Version 2 (HTTP/2)”, May 2015.

[24]. 3GPP TS29.500, " Technical Realization of Service Based Architecture; Stage 3", v.1.1.0, April 2018.

[25]. Vd. Sandeep Aggarwal1, Dr. Balbir Singh, A Case Study on Spinal Canal Stenosis – An Ayurvedic Prospective, TEJAS Journal of Technologies and Humanitarian Science, ISSN-2583-5599, Vol.02, I.03(2023)

[26]. 3GPP TS23.502, " Procedures for the 5G System; Stage 2", v.15.2.0, Jun. 2018.

[27]. ENISA, "Signaling Security in Telecom SS7/Diameter/5G," 2018. Imperva, " HTTP/2: In-depth analysis of the top four faws of the next generation web protocol", 2016

[28]. IETF RFC 7541, “HPACK: Header Compression for HTTP/2”, May 2015.

[29]. P. Patni, K. Iyer, R. Sarode, A. Mali, and A. Nimkar, "Man-in-the-middle attack in HTTP/2," presented at the 2017 International Conference on Intelligent Computing and Control (I2C2), 2017.

[30]. S. R. Hussain, O. Chowdhury, S. Mehnaz, and E. Bertino, "LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE," in Network and Distributed System Security Symposium, 2018.