**Uncovering the Path of an Online Request: To View Google’s Homepage, Type “https://www.google.com"**

The act of browsing the internet appears to be simple and easy. But retrieving and producing the requested web page requires several complex actions behind the scenes. The process of typing “https://www.google.com" into your browser and pressing Enter will be examined in this article, along with important details on DNS, TCP/IP, firewalls, HTTPS/SSL, load balancers, web servers, application servers, and databases.

The journey begins with a Domain Name System (DNS) request. Your browser sends a DNS query to a DNS server to resolve the domain name “[www.google.com](http://www.google.com/)" into an IP address. The DNS server then responds with the corresponding IP address of Google’s servers.

Once the IP address is obtained, the browser initiates a Transmission Control Protocol (TCP) connection to the server identified by that IP address. TCP ensures reliable, ordered, and error-checked delivery of data packets over the network. Internet Protocol (IP) handles addressing and routing of these packets. The TCP connection passes through any firewalls or network security measures that may be in place. Firewalls monitor and control incoming and outgoing network traffic based on predetermined security rules.

Secure Sockets Layer (SSL) or its successor, Transport Layer Security (TLS), encryption protocols are being utilised when the connection is made if the URL starts with “https://” (Hypertext Transfer Protocol Secure). Data sent between the browser and the server is encrypted and protected from tampering or eavesdropping thanks to SSL/TLS. Many high-traffic websites like Google employ load balancers to distribute incoming traffic across multiple servers. The load balancer acts as a traffic cop, routing requests to the least busy server to ensure optimal performance and reliability.

Once the request reaches the appropriate server, typically a web server like Apache, Nginx, or Microsoft IIS, it processes the request. The web server retrieves the requested web page or resources and prepares them for delivery. In some cases, the web server may need to interact with an application server to fulfil the request. Application servers, like Tomcat, Node.js, or Django, execute the business logic of web applications and generate dynamic content. If the requested web page or resources require data from a database, the application server interacts with the database server to retrieve the necessary information. Databases, such as MySQL, PostgreSQL, or MongoDB, store and manage structured data.

Finally, using the established TCP connection, the requested web page is built and delivered back to the browser together with any additional resources like images, scripts, and stylesheets. After that, the browser renders the content it has received, bringing up the well-known Google homepage.