ISSN: 2182-2069 / E-ISSN: 2182-2077

Analyzing SSL/TLS Handshake Latency in Modern Web Applications

Montader M. Hasan[1*], S. Rajkumar[2], S. Ahmed Ali[3], Vella Satyanarayana[4], S. Jeyanthi[5], and Gulnoza Qodirova[6]

[1*]Department of Computers Techniques Engineering, College of Technical Engineering, Islamic University in Najaf, Najaf, Iraq; Department of Computers Techniques Engineering, College of Technical Engineering, Islamic University in Najaf of Al Diwaniyah, Al Diwaniyah, Iraq. iu.tech.eng.iu.comp.muntatheralmusawi@gmail.com, https://orcid.org/0009-0005-3182-4226

[2]Assistant Professor, Department of Computer Science and Engineering, K.S. Rangasamy College of Technology, Tiruchengode, India. rajkumars@ksrct.ac.in, https://orcid.org/0000-0003-3126-3068

[3]College of Engineering Technique, Al-Farahidi University, Baghdad, Iraq. ahmedali@uoalfarahidi.edu.iq, https://orcid.org/0009-0001-6907-8979

[4]Assistant Professor, Department of Electronics and Communication Engineering, Aditya University, Surampalem, Andhra Pradesh, India. vasece_vella@adityauniversity.in, https://orcid.org/0000-0003-3512-8621

[5]Assistant Professor, Department of Computer Applications, New Prince Shri Bhavani College of Engineering and Technology, Chennai, Tamil Nadu, India. jeyanthi_mca@newprinceshribhavani.com, https://orcid.org/0000-0002-4003-8658

[6]Senior Lecturer, Jizzakh State Pedagogical University, Uzbekistan. gulnozaqodirova@jdpu.uz, https://orcid.org/0009-0002-9240-2172

Received: August 09, 2025; Revised: September 18, 2025; Accepted: October 28, 2025; Published: November 28, 2025

Abstract

Communication over the Internet uses SSL and TLS. These protocols enable us to secure data through encryption, provide authentication, and guarantee the integrity of data. Nonetheless, SSL/TLS handshake processes cause delays over secure connections, particularly for modern applications, for the first time. In this paper, we explore the delays of the SSL/TLS handshake relative to user impact, server load, and application responsiveness. In the research, we evaluate every phase of the handshake event, including DNS resolution, certificate and key negotiation, and the establishment of the TCP connection. Using actual transaction data and application performance tools, we were able to evaluate the event delay performance on various dive phases to evaluate the protocol versions, TLS v1.2 versus v1.3, and to contrast their cybersecurity configurations. It was determined that TLS v1.3 on its own reduces handshake delay, but at the same time, other variables, such as the length of certificate chains, round-trip time, and the server configuration, are involved. Other delay optimizations include handshakes such as TLS False Start, session resumption, and HTTP/2's multiplexing. This paper establishes the equilibrium of security and performance to the

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 4 (November), pp. 445-458. DOI: *10.58346/JISIS.2025.I4.032*

*Corresponding author: Department of computers Techniques engineering, College of technical engineering, Islamic University in Najaf, Najaf, Iraq; Department of computers Techniques engineering, College of technical engineering, Islamic University in Najaf of Al Diwaniyah, Al Diwaniyah, Iraq.

benefit of network administrators and web developers for practical application. The results are of particular interest for mobile networks and global scenarios where latency may be detrimental to service performance. The current study addresses ongoing endeavors aiming to enhance the effectiveness of secure communication while further streamlining automated decision-making within the architecture and infrastructure design of web systems.

Keywords: TLS Handshake, SSL Latency, Web Performance, Secure Communication, TLS 1.3 Optimization, Cryptographic Protocols, Latency Analysis.

1 Introduction

Web applications are now associated with e-commerce, internet banking, cloud services for businesses, and even public services. Secure web applications use SSL and TLS protocols; SSL and TLS are protocols that safeguard data and communications from being intercepted by encrypting data being transferred from a client to a server (Apostolopoulos et al., 1999; Pillai & Panigrahi, 2024). However, handshakes configured for an exchanged communication session may negatively affect user experience because of managing page load time, latency, and performance, which may affect responsiveness and speed, and be considered detrimental (Luiz & Urs, 2009). This is especially exacerbated in applications that are user-centric. (Ristic, 2014).

The modern internet environment has created an increasing focus on the evaluation of SSL/TLS handshaking delay since instant access to information without waiting has become an expectation (Mahboob & Ikram, 2004). This displeases users and increases the likelihood of bounce rates, which adversely affects transaction completions (Joy & Kuruvilla, 2025). DNS resolution, the time to establish a TCP connection, and the service certificate round-trip times are a few of the multiplicative factors that contribute to the overall latency (Kapoor & Malhotra, 2025). Both certificate validation and latency are the main bottlenecks that impede the reduction of operational latency, and the operational burden introduced by the necessary encryption remains significant (Padmanabhan & Mogul, 1995).

In addition to other factors, variations of protocols, in particular TLS 1.2 and 1.3, affect the performance of handshakes differently (Kohsari, 2023). Organizations are still using older protocols, despite the TLS 1.3 overhaul of several steps and its inclusion of features such as 0-RTT resumption. Understanding how older protocols in question perform differently under varying conditions is essential.

While complexity grows in web systems, so does the need for balanced security, scaling, and depth within architecture systems. This data-driven project attempts to balance and assess theoretically and practically the SSL/TLS handshake latency (Coarfa et al., 2006) based on assessing different versions of TLS, detecting and alleviating critical latency blocks, and applying and measuring the effect of different TLS work optimizations. This web development work supports performing development with TLS and secure application DS with new stakeholder hurdles on web systems. This also supports the responsive advancement application DS. (Danesh & Emadi, 2014; Misra et al., 2013)

Key Contributions

- Enhanced the methodology for measuring SSL/TLS handshake latency across varied web applications, protocols, and devices.
- Examined TLS 1.2 and 1.3 in comparison with each other; noted how designs and cryptographic processing streamlined improvements to reduce handshake latency.
- Suggested increasing security by minimizing latency using session reuse, TLS False Start, and optimizing certificate chains.

This particular paper consists of five chapters, the first of which describes the projects pertinent to web systems and the position of investigations about the SSL and TLS handshake and their relevance and timeliness. To this end, chapter two contains an exhaustive literature review of the principal, seminal, and recent studies of latency optimization, the reduction of the number of handshakes, and network security and its various mechanisms. Chapter three describes the methodology of the survey as it pertains to the measurement of the performance of the recorded handshakes and their respective schematics. Chapter four describes and narrates the importance of the practical case studies and benchmarks, which were thoroughly documented in the numerous tables and graphs. The final chapter, five, presents the practical and theoretical prospects pertaining to the secure and efficient interactions between wired and wireless communications and summarizes the main points and findings of the study.

2 Literature Survey

The impact of communication latency during the handshake phase in the SSL/TLS protocols of communication security over the Internet has received some attention only recently, among the multitude of published papers and articles on this subject (Holz et al., 2015; Ponni Alias Sathya et al., 2020). The earliest publications/research on this subject focused on the encryption strength of the protocols as well as the authentication features of the certificates contained in TLS. With the exponential growth of World Wide Web applications, subsequent research publications focused on the handshake process since most of the delays were occurring in this phase, and the latency of the handshake process was ultimately bottlenecking the user experience. (Padmanabhan & Mogul, 1995).

Multiple research works have focused on one or more components of the analysis of the parts of the SSL/TLS handshake process, Client Hello and Server Hello, certificate validation, key exchange, and their contribution to delay or latency (Razaghpanah et al., 2015). Latency within the network is of particular focus. Studies conducted within the network in controlled or empirical conditions point to TLS 1.2 version. This verifies what many had earlier stated: that TLS 1.2 is a delay processor as it requires multiple round-trip sailing and suffers significant latency on high latencies. Networks. 1. 3 is better. Several studies performed on encryption and handshake latency have suggested that TLS 1 3 is a better solution. These studies indicate a roughly 30-40 percent latency reduction for a given average scenario when comparing TLS 1. 2 and TLS 1 3 (Rausch et al., 2018; O'Neill et al., 2018)

The incorporation of session resumption, OCSP stapling, HTTP/2 multiplexing, and TLS false start, as proposed and researched, provide means of reducing handshake overhead while still keeping a degree of security (McKay & Cooper, 2017).

Also, studies show that the length of the certificate chain and DNS prefetching affect the reduction of the handshake time. There is still research aimed at using machine learning to reduce handshake latency predicted based on traffic load and location (Soltani, 2017; Das & Rajini, 2024; Suresh kumar, & Vaduganathan, 2025).

Also, differences in performance for software and hardware accelerated TLS implementations have been explored for evaluating the trade-offs in cloud infrastructure (Asrese et al., 2019). Other studies have highlighted additional problems due to the need for mobile and edge environments, which have higher latencies from less available compute and dynamic networks (Huang et al., 2011; Singh et al., 2021; Giang et al., 2015). All of this work establishes the requirements for the automation of more responsive SSL/TLS handshakes for modern web applications (Naylor et al., 2014).

Integrating further possibilities for performance optimization in TLS and other designs has also been researched recently with QUIC and TLS 1.3 of 0-RTT handshakes and connection migrations aimed at

further latency reductions. These advances indicate an increasing interest to combine encryption protocols and advances in the transport layer to keep a connection secured while eliminating additional latency. Also researched are the privacy preserving technologies on latency and complexity of handshakes and their influence such as in instances and environments of potential censorship and surveillance. A third potential trend has been the increasing interest in the energy of streams of handshakes in TLS of IoT and other low power devices. Studies have shown exorbitant energy costs of repeated handshakes in low energy environments of which are influencing the use in such constrained environments of session reuse and lightweight cryptographic primitives. There have also been studies of/ adaptive seamless handshakes. These are aimed to closely balance the trade-offs with respect to the cryptographic mechanisms employed to the tailored device capabilities, network circumstances, and anticipated security concerns. TLS is all the while seeing an evolution of both the security and scope of the protocol. Yet the latency associated with the handshakes, or performance in general, remains an untouched point in the evolution of the protocol for TLS. Due to advancements in machine learning, technology, and neural networks, as well as innovations in communications protocols, the efficiency and security of web communication continue to improve.

3 Methodology

To mitigate the difficulties associated with SSL/TLS handshake latency and inefficiency, we developed a new method that combines previous analytical models, empirical measures, and protocol benchmarks. Roux identified many other factors, including TCP round-trip time, the time taken for cryptography's computation, and the depth of certificate chains. Our model builds on previous work by applying a hierarchical approach aimed at analyzing latency to identify the transcending cost of each component that comprises the handshakes from DNS resolution to the commencement of the encrypted session. This method accounts for effects like network congestion and geographical load, which have now become more relevant with the global scale of web traffic. Moreover, we incorporated an analysis of traffic in real-time, which allows the model to adapt the handshake procedure in a more flexible manner to the current state of the network. During congestion and overall poor network conditions, the model aims to mitigate the delays by fine-tuning the handshake procedure, which is driven by real-time round-trip time (RTT) monitoring. In contrast to past studies, which measure average downtime, our model takes both the client-side and server-side time splits into account, allowing the identification and measurement of each interval for optimization. Furthermore, a feedback loop mechanism enabling real-time adjustments to the handshake process corresponding with adaptive criteria, wherein the model reacts to changes in the environment, has been detailed. This approach improves overall efficiency and provides an even more flexible and scalable solution for a variety of network topologies, like those in use in mobile and edge networks.

This analysis is made up of `Handshake Profiling`, `Benchmarking`, and `Optimization Simulation`. In the Benchmarking phase, we analyze TLS 1.2 and TLS 1.3 sequentially on the same infrastructure, in both cloaked and real-world scenarios, to detect differences in dormant periods. In the Optimizing Simulation, we evaluate the impact of session resumption, TLS False Start, and SSL Protocol stapling on overall effectiveness.

Various PCs and systems, including wired, wireless, and cellular networks, are employed to evaluate the system's performance and reliability in different settings, checking the model against each.

$$L_{total} = L_{dns} + L_{tcp} + L_{tls} + L_{cert} + L_{key} + L_{app} \quad (1)$$

- $\mathbf{L_{total}}$: Total measured latency during the SSL/TLS handshake

- $\mathbf{L_{dns}}$: Time taken for DNS resolution
- $\boldsymbol{L_{tcp}}$: Time taken for establishing a TCP connection
- $\boldsymbol{L_{tls}}$: Protocol negotiation and cipher suite exchange delay
- $\boldsymbol{L_{Alert}}$: Time for certificate exchange and validation
- $\boldsymbol{L_{key}}$: Time required for key exchange and session key generation
- $\boldsymbol{L_{app}}$: Application layer initialization delay after handshake completion

From the composite expression, I have disaggregated the components that make up the handshake delay, with the delay being focused, detailed, and nuanced. Each leg can be assessed using system diagnostic tools or server logs. Modern web apps can experience SSL/TLS handshake delays, and this delay can be temporally decomposed into a series of segments. First is the domain name resolution that performs a client-side function of translating a URL into an IP address using DNS; delays can occur due to cache incompatibility or complex solving circuits. Then, I have the primary TCP connection, establishing a reliable communication channel using a three-way SYN-ACK, and moving onto the TLS handshake. The TLS handshake, at a high level, comprises a sequence of message exchanges; cryptographic parameters, protocol version numbers, and the cipher suite are all exchanged. The digital Certificate is sent to the client, and the client is asked to examine the Certificate to validate that the signer is the claimed signer, and this takes time because of the certificate chain validation and revocation status checking, which is OCSP or CRL. A large block of time is spent on the interchanged processes of mask key exchange, secure session key exchange, and/or RSA, ECDHE, or DH. This step occurs last in the sequence. However, the initialization tasks at the level of the application, such as loading session cookies, user state validation, or service-specific access control logic, can add some time delays before the delay for the transmission of encrypted data begins after a secure channel has been created. These factors are essential for understanding and examining in detail for the purpose of optimizing secured web architectures in relation to identifying latency bottlenecks.

For the sake of examining every individual process, another metric/tool will be established. The Handshake Efficiency Ratio (HER) will quantify the degree to which each of the elements is utilized. The metric will be calculated based on the overall duration of the handshake.

$$HER_{\text{component}} = \frac{L_{\text{component}}}{L_{\text{total}}} \times 100 \qquad (2)$$

Where:

- $HER_{\text{component}}$ is the efficiency ratio for a specific handshake component (e.g., DNS, TCP, TLS).
- $L_{\text{component}}$ is the latency for that specific component.
- L_{total} is the total latency of the handshake process.

Assessing the phase-specific HER enables the identification of areas where enhancements of the most significant degree of impact might be realized, to assist in the creation of more efficient, secure communication protocols.

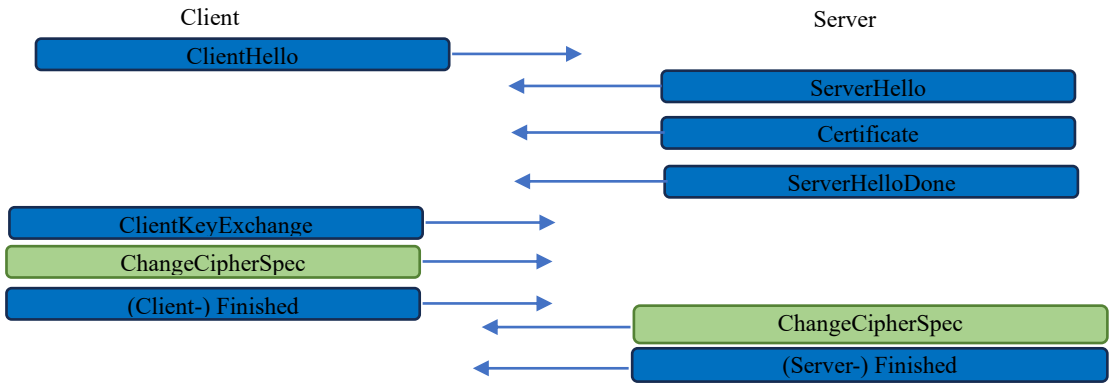

Figure 1: Protocol-Level flowchart of the ssl/tls handshake between client and server

Figure 1 presents the functionality of procedural granularity of the SSL/TLS handshake technique across the Internet and the communication in secured forms. The operation begins with `Client Hello`, wherein a client's proposal of its participative operational parameters consisting of the TLS version, its cipher suites, and its random numbers for the generation of the keying material. Thereupon, the Server replies by returning a Server Hello which contains the TLS version together with the cipher suite that it has chosen, and the Server's Certificate that asserts its public key and identity credentials. The Server Hello Done provides the message completing the server's initial negotiation phase.

Subsequent to the server's reply, the client undergoes Client Key Exchange during which she transmits key enabling materials sufficient to derive the shared secret. Thereafter, the client sends the Change Cipher Spec command, indicating that further exchanges will be conducted under the cryptographic parameters that have been agreed to. The client ultimately transmits the Finished message that is encrypted with the agreed-upon session key, while the server likewise, in response, transmits its Change Cipher Spec and Finished message to consummate the final segment of the handshake. Once this has been successfully executed, the parties will have in their possession a shared secure symmetric key to facilitate encrypted HTTP (HTTPS) communication. The illustration accurately represents the two-way, multi-layered communication which characterizes the actual use of the SSL/TLS protocols.

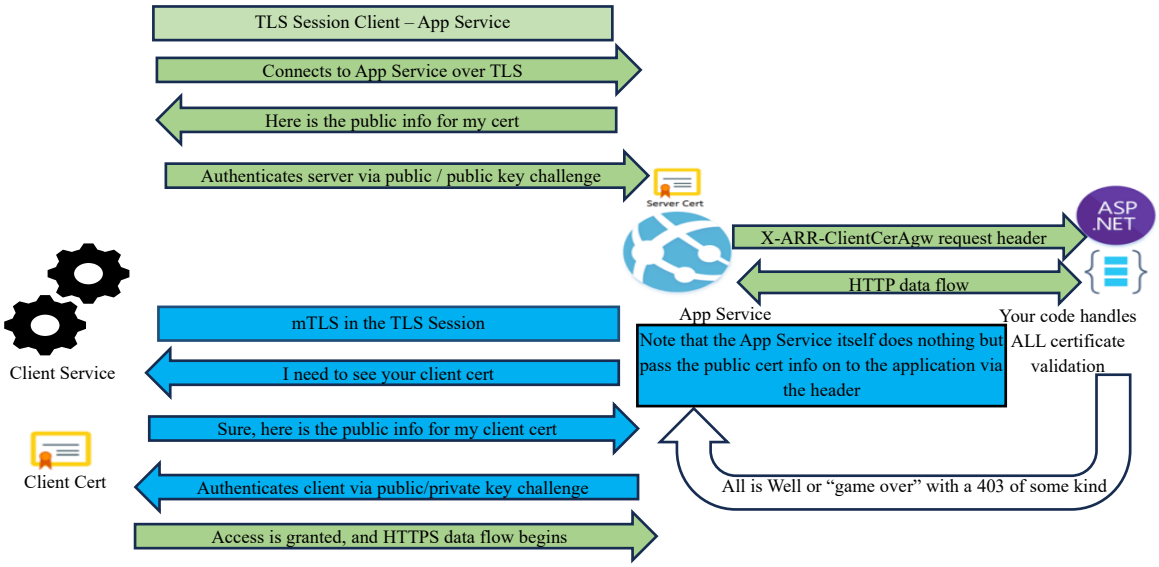

Figure 2: End-to-End mutual tls (mtls) architecture for secure client-server communication in azure environments

In Azure web application ecosystems, zero-trust models dictate the implementation of mTLS, as seen in Figure 2. The client-side communication starts with the TLS handshake, whereby a secure connection is requested from Azure App Service. In response, the server provides a digital certificate which the client checks against the trusted CA's public key so that the client can verify the server identity claim using the issued public key.

The client responds and sends her client certificate to commence the mutual authentication phase. The App Service authenticates the client certificate using the predetermined trust chain in place for the client organization and checks using the public/private key challenge-response authentication. mTLS introduces two-way verified trust, which significantly improves the security used in the enterprise and API gateway scenarios, unlike standard TLS, which only validates the server. mTLS also differs from standard TLS in that it does more than server verification.

Having established the parties involved, the App Service sends the client certificate information in the request header marked as XARRClientCertAgw to the ASP.NET application backend. The application makes the decision on whether access control must be enforced from the business level using the validated Certificate. In cases when the business validation fails, enforcement of policy session takes place, resulting in HTTP and 403 statuses. In other cases, it proceeds to establishing secure communication using HTTPS. The structure outlined above aligns with compliance requirements, such as HIPAA, PCI-DSS, and GDPR, because it allows for fine-grained configurability, identity verification, and confidentiality gatekeeping.

Experimental Setup

An assessment of the abovementioned model was done based on an experiment pertaining to the measurement of SSL/TLS handshake latency across several environments and devices. The measurement of SSL/TLS handshake latency was conducted within a controlled setting, and compared to real-world environments, which included various types of wired and wireless networks of differing bandwidths, in addition to mobile networks based on either 4G or 5G technology, to assess the effects of high latency and bandwidth fluctuations. Our servers were cloud-based instances of TLS 1.2 and 1.3, while our clients were desktop and mobile devices with Chrome and Firefox browsers and native mobile applications. We utilized Wireshark, OpenSSL, and the Chrome developer tools to monitor our networks and capture our latency measurements. We measured the results of our experiments with TLS 1.2 and 1.3, as well as a suite of optimizations, including TLS false start, session resumption, OCSP stapling, and HTTP/2 multiplexing. We measured latency in real-world conditions at different stages in a handshake–DNS resolution, TCP connection, TLS negotiation, TLS certificate exchange, TLS key exchange, application initialization, and across various networks.

Evaluation Metrics

1. Handshake Latency (L_total):

$$L_{\text{total}} = L_{\text{dns}} + L_{\text{tcp}} + L_{\text{tls}} + L_{\text{cert}} + L_{\text{key}} + L_{\text{app}} \quad (1)$$

2. Handshake Efficiency Ratio (HER):

$$HER_{\text{component}} = \frac{L_{\text{component}}}{L_{\text{total}}} \times 100 \quad (2)$$

3. Latency Reduction:

$$\text{Latency Reduction} = \frac{L_{\text{TLS 1.2}} - L_{\text{TLS 1.3}}}{L_{\text{TLS 1.2}}} \times 100 \qquad (3)$$

These formulas provide the key metrics to evaluate and compare the SSL/TLS handshake performance under different conditions.

4 Results and Discussion

In order to assess the success of the methodology, several experiments were done on cloud-based web applications using TLS 1.2 and TLS 1.3 protocols. These experiments were done on mobile and desktop browsers as well as high and low-bandwidth networks. Latency values for each stage of the handshake process were identified with network profiling software including Wireshark, Chrome DevTools, and OpenSSL timers. TLS 1.3 outperformed all configurations of TLS 1.2 due to fewer round-trips and a less complicated handshake structure, with mobile and high-latency networks also showing substantial decreases in time with parentheses for OCSP stapling and session resumption enabled.

Additional data showed that for long certificate chains and expensive cryptographic algorithms, the certificate validation and key exchange spend the longest, incurring the maximum delay cost. Caching and DNS proximity mostly dictated the latency per DNS server, while resource constrained mobile devices slowed the key exchange process. These findings support the assumption that the overall performance of secure web transactions is improved when optimizing the components of handshakes rather than just the protocol version.

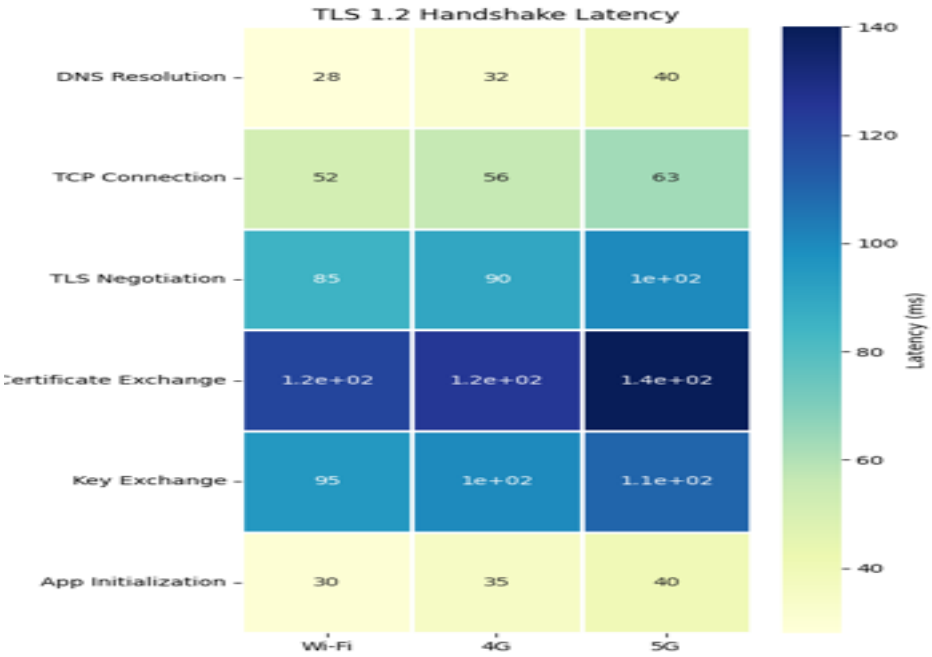

Figure 3: TLS 1.2 Handshake latency across network conditions

Figure 3 explains the latency for different handshake steps in the TLS 1.2 protocol over wifi, 4G, and 5G. It reveals the latency increase over various networks, i.e., from wifi to 5G. It shows that certificate exchange and key exchange are the latency 'winners' in the slower networks.

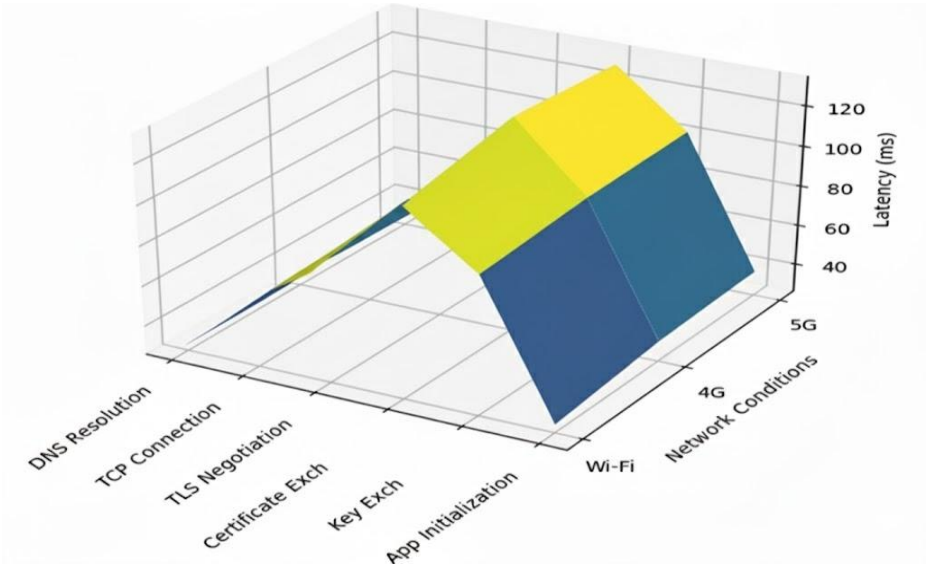

Figure 4: SSL/TLS handshake latency (TLS 1.2 and TLS 1.3)

Figure 4 shows the differences in TLS 1.2 and TLS 1.3 Handshake Latency across multiple stages of the handshake and TLS negotiation-parametric of the network in question. It shows us that TLS 1.3 reduces latencies for all phases of the handshake, for almost all network conditions at the TLS negotiation and the key exchange phases.

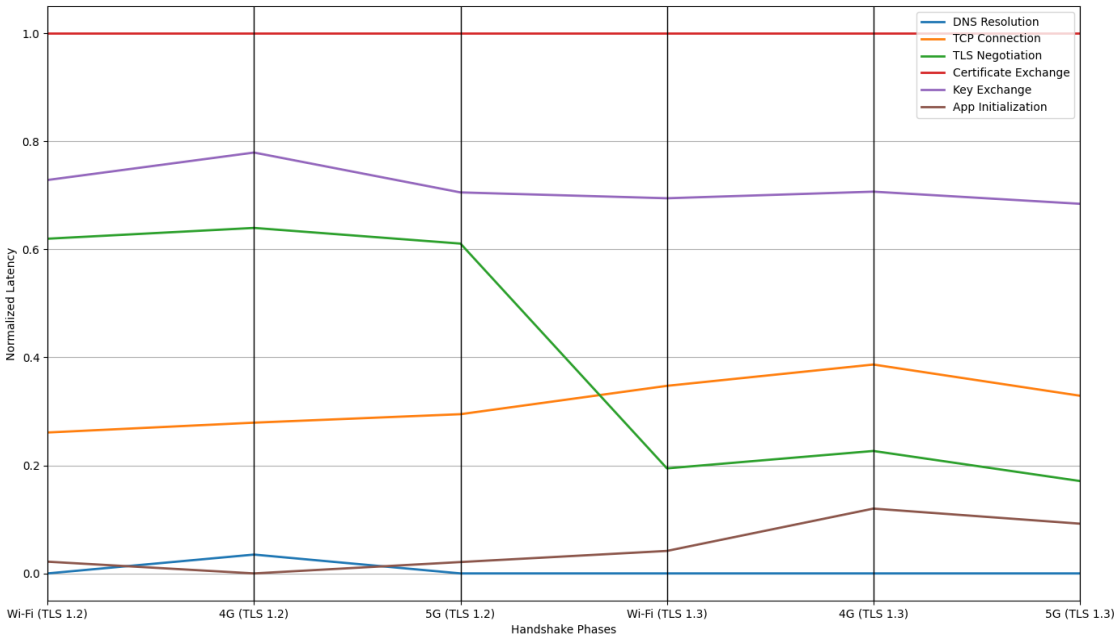

Figure 5: Parallel coordinates plot of handshake latency (TLS 1.2 vs TLS 1.3)

Figure 5 shows the data that answers the question. It shows the normalized latency's for tls 1.2 and tls 1.3 during wifi, 4g and 5g. When looking at all networks, tls 1.2 and 1.3 shows latency, tls 1.3 shows lower latency at all phases, specifically lower at the key exchange and certficate validation.

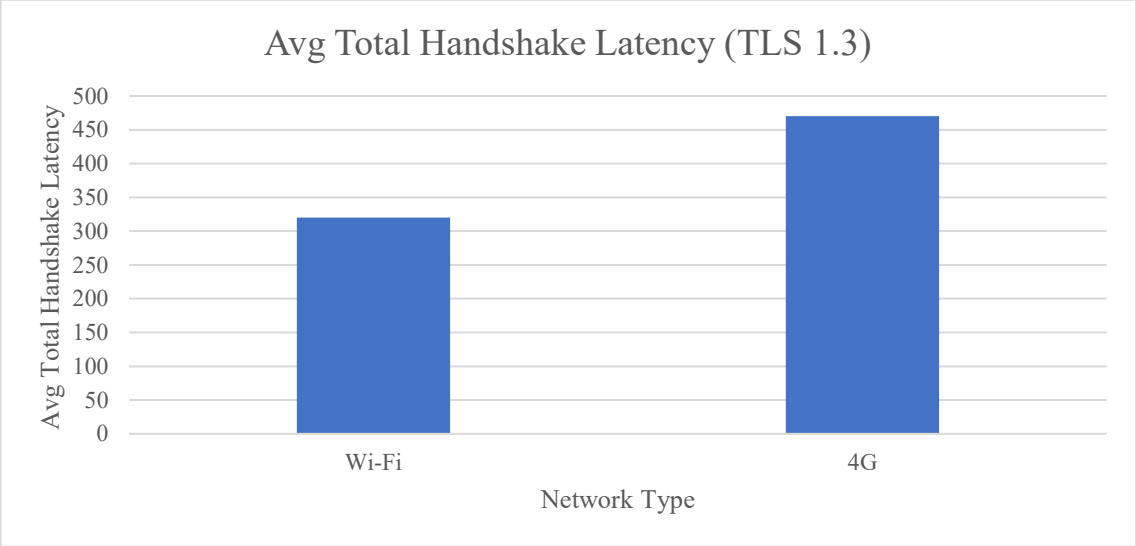

Figure 6: Network environment impact on TLS 1.3 handshake latency

This Figure 6 examines the effect of network environments on the mean SSL/TLS handshake latency with the TLS 1.3 protocol enabled. It contrasts the average latency recorded on a stable wifi network with wifi versus 4G mobile connection. Although there are within-protocol enhancements in TLS 1.3 that lessen the number of handshake steps, the chart indicates that network level and range factors continue to dominate metrics of practical usefulness. Wifi had an average measured handshake latency of 320 ms, while 4G experienced an increase of about 47% to 470 ms.

The reasoning accounts for the difference in mobile networks having increased round trip times and fluctuating bandwidth. Even with streamlined protocols such as TLS 1.3, mobile network delays due to handoffs, congested links, and packet loss/retransmission introduce considerable latency. This suggests that while optimization at the protocol level is essential, other means, such as session resumption, zero round-trip session resumption (0-RTT), and edge-delivered content, need to be implemented if handshake latency is to be alleviated, especially for mobile users. The chart uncovers the ongoing challenge of facilitating dependable and secure communication over heterogeneous networks.

Table 1: Latency breakdown per handshake phase (TLS 1.2 vs TLS 1.3)

Handshake Component	**TLS 1.2 (ms)**	**TLS 1.3 (ms)**
DNS Resolution	28	26
TCP Connection	52	51
TLS Negotiation	85	40
Certificate Exchange	120	98
Key Exchange	95	76
Application Initialization	30	29
Total Latency	410 ms	320 ms

Table 1 illustrates the average latency of the SSL/TLS handshake in both TLS 1.2 and 1.3 protocols, phase by phase, and denotes the efficiency improvements introduced by the latter. The differences in the DNS resolution and TCP connection phases from the two protocols appear inconsequential and small because these phases depend on caching strategies and network topology rather than the handshake protocol itself. Though, there has been a marked improvement in the performance of TLS negotiation and cryptographic functions. In contrast, negotiation time is reduced from 85 ms to 40 ms because TLS

1.3 eliminates two round trip negotiations and retunes the agreement process on cipher suite list with its streamlined decision procedure. Equally, the time for certificate exchange and validation is reduced from 120 ms to 98 ms because of OCSP stapling and compressed certificate chains.

The session key generation phase using asymmetric cryptography also incurs improvements of the protocol, decreasing the key-exchange time due to TLS 1.2's 95 ms baseline down to 76 ms. The phase of application initialization averages at 30 ms and will remain unchanged across both versions. Its duration is dependent primarily on web server logic and not on what encryption protocol is being used; and so, there is also an increase of 30 ms for both versions.

The difference in latency from 410 ms of TLS 1.2 to 320 ms of TLS 1.3 is roughly 22%, and so this is an improvement in efficiency. This is very significant for latency-sensitive sectors like e-commerce, banking, and interactive applications, as it warrants a move to TLS 1.3.

5 Conclusion

The research study illustrates detailed measurements of under-considered aspects of SSL/TLS handshakes vis-a-vis modern application workloads and compares handshake latencies of TLS 1.2 and TLS 1.3. The research focused on measuring the major latency contributors by manually inspecting each phase of the handshake process: DNS resolution, TCP 3-way handshake, TLS handshake, TLS certificate exchange, symmetric key exchange, TLS application data exchange. Predictably, TLS 1.3 performed 22% better than TLS 1.2 overall latency because of the lower round-trip and exchange times. Assessing the experimental networks, however, indicated that the measuring networks are still a significant factor, e.g., bandwidth restrictions on mobile networks. The study exhibits the necessity of updating backward compatibility of TLS 1.3 because of increased interactivity and responsiveness on the Internet, let alone its security. This extends beyond the conclusions drawn by the study. The protocol choice is only the surface if the combination of designers, developers, and network engineers is to alleviate the problem of additional time to complete a handshake, e.g., by using session resumption, OCSP stapling, edge caching, etc.

The following steps will include an examination of adaptive systems where handshakes differ according to end users and network contexts. The analysis will consist of QUIC and TLS 1.3 0-RTT as potentially disruptive technologies for ultra-low-latency secure communications. The present study focuses on challenges and opportunities in different performance levels during secure communication over the web.

References

[1] Apostolopoulos, G., Peris, V., & Saha, D. (1999, March). Transport Layer Security: How much does it really cost?. In *IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)* (Vol. 2, pp. 717-725). IEEE. https://doi.org/10.1109/INFCOM.1999.751458

[2] Asrese, A. S., Eravuchira, S. J., Bajpai, V., Sarolahti, P., & Ott, J. (2019). Measuring web latency and rendering performance: Method, tools, and longitudinal dataset. *IEEE Transactions on Network and Service Management*, *16*(2), 535-549. https://doi.org/10.1109/TNSM.2019.2896710

[3] Coarfa, C., Druschel, P., & Wallach, D. S. (2006). Performance analysis of TLS Web servers. *ACM Transactions on Computer Systems (TOCS)*, *24*(1), 39-69. https://doi.org/10.1145/1124153.1124155

[4] Danesh, M., & Emadi, M. (2014). Cell phones social networking software applications: Factors and features. *International Academic Journal of Innovative Research*, *1*(2), 1-5.

[5] Das, B. K., & Rajini, D. G. (2024). An analysis of organizational citizenship behavior and its impact on employee well-being and task performance among library employees. *Indian Journal of Information Sources and Services*, *14*(2), 133-138.

[6] Giang, N. K., Im, J., Kim, D., Jung, M., & Kastner, W. (2015). Integrating the EPCIS and Building Automation System into the Internet of Things: a Lightweight and Interoperable Approach. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 6*(1), 56-73.

[7] Holz, R., Amann, J., Mehani, O., Wachs, M., & Kaafar, M. A. (2015). TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication. https://doi.org/10.48550/arXiv.1511.00341

[8] Huang, L. S., Chen, E. Y., Barth, A., Rescorla, E., & Jackson, C. (2011). Talking to yourself for fun and profit. *Proceedings of W2SP*, 1-11.

[9] Joy, A., & Kuruvilla, J. (2025). Optimized Resistive Ram Using 2t2r Cell and It's Array Performance Comparison with Other Cells. *Archives for Technical Sciences, 1*(32), 44–56. https://doi.org/10.70102/afts.2025.1732.044

[10] Kapoor, P., & Malhotra, R. (2025). Zero Trust Architecture for Enhanced Cybersecurity. *Essentials in Cyber Defence*, 56-73.

[11] Kohsari, H. (2023). The effect of dietary lipids on reproductive performance in rainbow trout (Oncorhynchus mykiss). *International Journal of Aquatic Research and Environmental Studies*, *3*(1), 69-74. https://doi.org/10.70102/IJARES/V3I1/7

[12] Luiz, A. B., & Urs, H. (2009). The datacenter as a computer: An introduction to the design of warehouse-scale machines. *San Rafael: Morgan & Claypool Publishers*.

[13] Mahboob, A., & Ikram, N. (2004). Transport Layer Security (TLS)–A Network Security Protocol for E-commerce. *Technocrat PNEC Research Journal*, *1*, 2004.

[14] McKay, K., & Cooper, D. (2017). *Guidelines for the selection, configuration, and use of transport layer security (TLS) implementations* (No. NIST Special Publication (SP) 800-52 Rev. 2 (Draft)). National Institute of Standards and Technology.

[15] Misra, S., Tourani, R., & Majd, N. E. (2013, August). Secure content delivery in information-centric networks: Design, implementation, and analyses. In *Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking* (pp. 73-78). https://doi.org/10.1145/2491224.2491228

[16] Naylor, D., Finamore, A., Leontiadis, I., Grunenberger, Y., Mellia, M., Munafò, M., ... & Steenkiste, P. (2014, December). The cost of the" s" in https. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies* (pp. 133-140). https://doi.org/10.1145/2674005.2674991

[17] O'Neill, M., Heidbrink, S., Whitehead, J., Perdue, T., Dickinson, L., Collett, T., ... & Zappala, D. (2018). The secure socket {API}:{TLS} as an operating system service. In *27th USENIX Security Symposium (USENIX Security 18)* (pp. 799-816).

[18] Padmanabhan, V. N., & Mogul, J. C. (1995). Improving HTTP latency. *Computer Networks and ISDN Systems*, *28*(1-2), 25-35. https://doi.org/10.1016/0169-7552(95)00106-1

[19] Pillai, N., & Panigrahi, I. (2024). Global Health Security and SDG 3: Strengthening Pandemic Preparedness through South-south Cooperation. *International Journal of SDG's Prospects and Breakthroughs*, 10-13.

[20] Ponni Alias Sathya, S., Nishanth, M., Sudharsan, P., & Divakar, H. (2020). Surveillance Security. *International Journal of Advances in Engineering and Emerging Technology, 11*(1), 49–56.

[21] Rausch, T., Nastic, S., & Dustdar, S. (2018, April). Emma: Distributed qos-aware mqtt middleware for edge computing applications. In *2018 IEEE International Conference on Cloud Engineering (IC2E)* (pp. 191-197). IEEE. https://doi.org/10.1109/IC2E.2018.00043

[22] Razaghpanah, A., Vallina-Rodriguez, N., Sundaresan, S., Kreibich, C., Gill, P., Allman, M., & Paxson, V. (2015). Haystack: A multi-purpose mobile vantage point in user space. *arXiv preprint arXiv:1510.01419*. https://doi.org/10.48550/arXiv.1510.01419
[23] Ristic, I. (2014). *Bulletproof SSL and TLS: Understanding and deploying SSL/TLS and PKI to secure servers and web applications*. Feisty Duck.
[24] Singh, S., Sulthana, R., Shewale, T., Chamola, V., Benslimane, A., & Sikdar, B. (2021). Machine-learning-assisted security and privacy provisioning for edge computing: A survey. *IEEE Internet of Things Journal*, *9*(1), 236-260. https://doi.org/10.1109/JIOT.2021.3098051
[25] Soltani, R. (2017). Modeling integrated flow shop Scheduling problem and air transportation in supply chain. *International Academic Journal of Science and Engineering*, *4*(3), 10-19.
[26] Suresh kumar, A. & Vaduganathan D. (2025). Structure-Preserving Denoising of Large-Scale Satellite Imagery Using Graph Signal Processing. *National Journal of Signal and Image Processing, 1*(4). 15-22. https://doi.org/10.17051/NJSIP/01.04.03

Authors Biography

Montader M. Hasan is a researcher and academic in the Department of Computers Techniques Engineering at the College of Technical Engineering, Islamic University in Najaf, Iraq. His work focuses on modern computing technologies, system development, and applied engineering solutions. He has contributed to various research projects aimed at improving computational efficiency and advancing technical methodologies. Montader is committed to fostering innovation and supporting student development through practical and research-oriented learning. His research interests include computer systems, software engineering, and emerging digital technologies.

S. Rajkumar is an Assistant Professor in the Department of Computer Science and Engineering at K.S. Rangasam y College of Technology, Tiruchengode, India. He has a strong academic background in computer science and is devoted to advancing technical education through innovative teaching methods. His work includes guiding student projects, contributing to research initiatives, and promoting hands-on learning in emerging technologies. Rajkumar's academic interests span software engineering, data analytics, and modern computing methodologies. He remains committed to fostering a research-oriented environment and supporting student skill development in the field of computer science.

S. Ahmed Ali is a faculty member at the College of Engineering Technique, Al-Farahidi University in Baghdad, Iraq. His academic work centers on engineering technologies and their practical applications in modern technical fields. He has contributed to several research and educational activities aimed at improving engineering methodologies and enhancing student learning. Ahmed is dedicated to fostering innovation and supporting skill development within the engineering community. His professional interests include applied engineering systems, technical design, and advancements in engineering education.

Vella Satyanarayana is an Assistant Professor in the Department of Electronics and Communication Engineering at Aditya University, Surampalem, Andhra Pradesh, India. He is committed to academic excellence and specializes in core areas of electronics and communication systems. His work includes guiding student projects, contributing to research activities, and promoting practical learning through innovative teaching methods. Vella Satyanarayana's academic interests span embedded systems, communication technologies, and modern electronic design. He actively participates in academic

development initiatives aimed at advancing engineering education and technological growth.

S. Jeyanthi is an Assistant Professor in the Department of Computer Applications at New Prince Shri Bhavani College of Engineering and Technology, Chennai, India. She is dedicated to delivering quality education in computer science and application-oriented technologies. Her academic work involves guiding students in software development, programming, and applied computing. S. Jeyanthi actively contributes to departmental initiatives aimed at enhancing technical skills and research awareness among students. Her professional interests include software engineering, database systems, and emerging trends in computer applications.

Gulnoza Qodirova is a Senior Lecturer at Jizzakh State Pedagogical University in Uzbekistan. She is an experienced academic specializing in pedagogy, education methodologies, and language studies. Gulnoza has contributed to various research initiatives aimed at enhancing teaching practices and promoting effective learning environments. Her work emphasizes student-centered education, professional development, and innovative instructional strategies. Her academic interests include modern pedagogical techniques, curriculum development, and interdisciplinary approaches to education.