International Journal of
Technology and
Systems
(IJTS)

Évaluation Expérimentale de l'impact du Chiffrement TLS sur les Performances Réseau : Comparaison entre HTTP et HTTPS en Environnement Contrôlé.

Journal of Technology and Systems
ISSN: 2788-6344 (Online)
Vol. 8, Issue No. 2, pp 29 – 43, 2026
www.carijournals.org

Évaluation Expérimentale de l'impact du Chiffrement TLS sur les Performances Réseau : Comparaison entre HTTP et HTTPS en Environnement Contrôlé.

[1*]Paluku Mukovi Victoire, [2]Paluku Tshiyire Odilon

[1]Sciences Informatique, Université Adventiste de Lukanga

[2]Managements des Systèmes d'Information, Université Adventiste de Lukanga

https://oricd.org/0009-0001-5475-7602

Accepted: 27th May, 2026, Received in Revised Form: 28th June, 2026, Published: 30th June, 2026

Résumé

Objectif : Cette étude évalue expérimentalement l'impact du chiffrement TLS sur les performances réseau en comparant HTTP et HTTPS en environnement contrôlé. Elle mesure l'effet du chiffrement sur la latence, le débit et la taille des données échangées.

Méthodologie : Une application Python développée avec Flet automatise les mesures. Les expérimentations ont été réalisées sur un serveur Apache local (XAMPP) hébergeant le même fichier HTML sous HTTP et HTTPS. Trente requêtes par protocole ont été exécutées. Les données ont été exportées en CSV puis analysées statistiquement. Les tests utilisés sont Shapiro-Wilk, test t apparié ou Wilcoxon selon la distribution, complétés par la taille d'effet.

Résultats : La latence est significativement plus faible sous HTTPS que HTTP ($p < 0,0001$; $d = 1,27$), tandis que le débit est significativement plus élevé sous HTTPS ($p < 0,0001$; $r = 0,86$). Aucune différence significative n'est observée pour la taille des données transférées ($p = 0,2325$). Ces résultats indiquent que TLS n'entraîne pas de dégradation des performances et peut même améliorer celles-ci grâce aux optimisations modernes.

Recommandations : Les résultats encouragent l'adoption de HTTPS garantissant sécurité sans perte de performance en environnement maîtrisé. Des travaux futurs devraient étendre l'étude à différents contenus, réseaux réels (Wi-Fi, 4G/5G) et protocoles HTTP/2, HTTP/3 et TLS 1.3.

Mots-clés : *TLS, Chiffrement, Performances Réseau, HTTP, HTTPS, Latence, Débit De Transmission, Sécurité Réseau, Environnement Contrôlé, Évaluation Expérimentale.*

Journal of Technology and Systems
ISSN: 2788-6344 (Online)
Vol. 8, Issue No. 2, pp 29 – 43, 2026
www.carijournals.org

Abstract

Purpose: This study experimentally evaluates the impact of TLS encryption on network performance by comparing HTTP and HTTPS in a controlled environment. It measures the effect of encryption on latency, throughput, and the size of exchanged data.

Methodology: A Python application developed with Flet was used to automate measurements. The experiments were conducted on a local Apache server (XAMPP) hosting the same HTML file under both HTTP and HTTPS. Thirty requests per protocol were executed. The data were exported in CSV format and then statistically analyzed. The tests used include the Shapiro-Wilk normality test, the paired t-test or Wilcoxon test depending on data distribution, complemented by effect size calculations.

Findings: Latency is significantly lower under HTTPS than HTTP ($p < 0.0001$; $d = 1.27$), while throughput is significantly higher under HTTPS ($p < 0.0001$; $r = 0.86$). No significant difference is observed for the size of transferred data ($p = 0.2325$). These results indicate that TLS does not degrade performance and may even improve it due to modern optimizations.

Recommendations: The results support the adoption of HTTPS, ensuring security without performance loss in a controlled environment. Future work should extend this study to different types of content, real-world networks (Wi-Fi, 4G/5G), and protocols such as HTTP/2, HTTP/3, and TLS 1.3.

Keywords: *TLS, Encryption, Network Performance, HTTP, HTTPS, Latency, Throughput, Network Security, Controlled Environment, Experimental Evaluation.*

1. Introduction

1.1.Cadre général de la communication web et enjeux

La communication web repose sur un modèle d'échange client–serveur dans lequel des systèmes distribués interagissent à travers des protocoles standardisés de la couche application. Ces protocoles définissent les règles de structuration, de transmission et de réception des données sur les réseaux informatiques, permettant ainsi l'interopérabilité entre environnements hétérogènes. Dans ce cadre, le Web constitue un espace de communication numérique dont les performances, la fiabilité et la sécurité sont des critères essentiels de qualité de service (Fielding et al., 2019).

L'un des enjeux majeurs de cette architecture réside dans la coexistence de deux exigences souvent antagonistes : la sécurité des communications et l'efficacité des performances réseau. En effet, les applications web modernes ne se limitent plus à la simple diffusion de contenu statique, mais supportent désormais des services critiques tels que les transactions financières, les systèmes de santé numériques et les infrastructures cloud. Ces usages exigent à la fois une faible latence et une garantie forte de confidentialité, d'intégrité et d'authentification des communications (Rescorla, 2018).

Dans cette perspective, les protocoles de communication jouent un rôle central dans la structuration des échanges sur Internet. Le protocole HTTP (HyperText Transfer Protocol) constitue historiquement le fondement des communications web, assurant le transfert de ressources selon un modèle requête–réponse. Toutefois, sa version initiale ne fournit aucun mécanisme natif de sécurité, ce qui expose les données transmises à des attaques telles que l'interception ou la modification en transit (Fielding et al., 2019).

Pour répondre à ces limitations, une évolution du protocole HTTP a été introduite sous la forme de HTTPS (HyperText Transfer Protocol Secure), qui repose sur l'intégration d'une couche de sécurité cryptographique assurée par TLS (Transport Layer Security). TLS permet d'établir un canal chiffré entre client et serveur, garantissant la confidentialité, l'intégrité et l'authenticité des communications (Rescorla, 2018). Dans ce protocole, la phase de handshake permet la négociation des paramètres cryptographiques et l'établissement sécurisé de la session.

Au-delà de la dimension sécuritaire, l'évolution des protocoles web est étroitement liée aux exigences de performance des systèmes distribués. Les temps de réponse, le débit de transmission et la latence sont devenus des indicateurs critiques de qualité de service, influençant directement l'expérience utilisateur et l'adoption des services numériques. Plusieurs travaux montrent que ces métriques peuvent être affectées par les mécanismes de sécurité introduits par TLS, notamment lors de l'établissement initial de la connexion (Dierks & Rescorla, 2008 ; Langley et al., 2017).

Par ailleurs, cette évolution soulève également des enjeux liés à l'impact environnemental des infrastructures numériques. En effet, les opérations cryptographiques associées aux protocoles sécurisés impliquent une consommation accrue de ressources de calcul, notamment au niveau des processeurs serveurs. À grande échelle, cette consommation contribue à l'empreinte

énergétique globale des systèmes numériques, ce qui place l'optimisation des protocoles web dans une perspective de sobriété énergétique des infrastructures informatiques (Andrae & Edler, 2015).

Dans ce contexte, l'étude des protocoles HTTP et HTTPS ne peut se limiter à une analyse fonctionnelle ou sécuritaire. Elle doit également intégrer une réflexion sur les compromis entre sécurité, performance et efficacité énergétique des systèmes de communication modernes. C'est dans cette perspective que s'inscrit la présente recherche.

1.2. Contexte

L'adoption des protocoles sécurisés dans les communications web a connu une croissance rapide au cours de la dernière décennie. Le protocole HTTPS (HyperText Transfer Protocol Secure), basé sur le chiffrement TLS (Transport Layer Security), est aujourd'hui largement déployé comme standard de sécurisation des échanges sur Internet. Selon le Google Transparency Report, plus de 95 % du trafic web sur le navigateur Chrome est désormais chiffré via HTTPS dans plusieurs régions du monde (Google, 2023).

Cette évolution est motivée par la nécessité de protéger les données des utilisateurs contre les interceptions et les attaques réseau, notamment dans les environnements ouverts. HTTPS est ainsi devenu une norme recommandée pour les sites web traitant des informations sensibles, en raison de ses garanties de confidentialité, d'intégrité et d'authentification des communications (Rescorla, 2018).

1.3. Problématique

Malgré les avantages sécuritaires offerts par HTTPS, son fonctionnement repose sur le protocole TLS, qui introduit des opérations cryptographiques supplémentaires telles que le handshake, le chiffrement et le déchiffrement des données. Ces mécanismes peuvent potentiellement influencer les performances réseau, notamment en termes de latence, de débit et de consommation des ressources système (Dierks & Rescorla, 2008 ; Langley et al., 2017). Dans ce contexte, une interrogation subsiste : Comment les variations de configurations et de paramètres sécuritaires du chiffrement TLS affectent-elles les performances réseaux lors de la transmission de données entre clients et serveurs ?

1.4. Hypothèse

Nous posons l'hypothèse générale que l'introduction du chiffrement TLS engendre un surcoût mesurable en matière de performance réseau, notamment au niveau de la latence et du débit.

Formellement, cette hypothèse se décline comme suit :

- Hypothèse nulle (H_0) : Il n'existe aucune différence significative entre HTTP et HTTPS en ce qui concerne la latence, le débit ou la taille des données échangées.
- Hypothèse alternative (H_1) : Il existe une différence significative entre HTTP et HTTPS sur au moins une des variables mesurées (latence, débit, ou taille).

Ces hypothèses seront testées à l'aide de méthodes statistiques adaptées aux conditions de distribution des données (normalité ou non).

1.5. Objectifs

Cette étude vise deux objectifs principaux :

1. Quantifier l'impact du chiffrement TLS sur des indicateurs clés de performance réseau tels que la latence, la taille des données échangées et le débit.
2. Comparer objectivement les performances de HTTP et HTTPS dans un environnement contrôlé où toutes les autres variables sont strictement maintenues constantes.

1.6. Méthodologie résumée

Pour atteindre ces objectifs, nous avons conçu une application expérimentale en Python à l'aide du framework Flet, permettant l'automatisation de requêtes HTTP/HTTPS vers un serveur Apache local configuré via XAMPP. Le même fichier est servi sous les deux protocoles, et les temps de réponse sont mesurés et enregistrés automatiquement dans un fichier CSV pour analyse. L'environnement de test est volontairement limité et reproductible afin de garantir la validité interne des résultats.

1.7. Plan de l'article

La suite de cet article est structurée comme suit : la section 2 présente le cadre théorique et les travaux antérieurs sur le sujet. La section 3 décrit en détail la méthodologie expérimentale mise en place. Les résultats sont présentés et analysés dans la section 4, puis interprétés dans la section 5. Enfin, la discussion et les limites de l'étude sont abordées dans la section 6, avant la conclusion et les perspectives futures.

2. Fondements théoriques et revue de la littérature

2.1. Protocoles de communication web

Le protocole HTTP (HyperText Transfer Protocol) est un protocole de la couche application utilisé pour la transmission de documents hypertextes sur le Web. Il fonctionne par défaut sur le port 80, sans chiffrer les données transmises. Par conséquent, les échanges entre client et serveur via HTTP sont vulnérables aux attaques de type *man-in-the-middle*, à l'interception ou à la modification des données (Fielding, et al., 2019).

En réponse à ces failles, le HTTPS (HTTP Secure) a été introduit. HTTPS est essentiellement HTTP encapsulé dans une couche de chiffrement sécurisée, généralement TLS (Transport Layer Security), fonctionnant par défaut sur le port 443. Grâce à TLS, HTTPS garantit l'authenticité, l'intégrité et la confidentialité des données (Rescorla, HTTP Over TLS. RFC 2818., 2000). Cette sécurité supplémentaire implique toutefois des opérations supplémentaires comme le handshake TLS, le chiffrement des données côté client, et leur déchiffrement côté serveur.

2.2. Le chiffrement TLS comme facteur de surcoût

Le protocole TLS introduit plusieurs étapes supplémentaires dans l'établissement de la connexion. L'une des plus coûteuses est le handshake TLS, une phase initiale durant laquelle

le client et le serveur s'échangent des clés cryptographiques, négocient la suite cryptographique (cipher suite) à utiliser, et établissent un canal sécurisé.

Chaque connexion HTTPS nécessite une série d'opérations cryptographiques :

- Génération d'une clé de session,
- Chiffrement des messages,
- Déchiffrement et vérification côté destinataire,
- Vérification de certificats et négociation de paramètres (Oppliger, 2009).

Ces opérations entraînent une consommation accrue de ressources CPU ainsi qu'une augmentation du temps de latence initial, surtout lors de la première connexion. Bien que TLS 1.3 ait amélioré ces performances par rapport à TLS 1.2, le coût reste non nul (Langley, Modadugu, & Moeller, The design of TLS 1.3. IETF Draft., 2017).

2.3. Mesures de performance réseau

L'évaluation des performances réseau repose sur plusieurs **métriques fondamentales** :

- **Latence** : Temps nécessaire pour qu'un paquet de données effectue un aller-retour entre le client et le serveur.
- **Bande passante** : Capacité maximale de transmission de données sur un canal donné, généralement exprimée en Mbps (mégabits par seconde).
- **Débit effectif** : Quantité réelle de données transmises avec succès par unité de temps, souvent mesurée en Ko/s ou Mo/s. Contrairement à la bande passante, le débit dépend de la latence, des erreurs de transmission, du protocole utilisé, etc.

Ces trois mesures sont directement impactées par le protocole de communication choisi, surtout lorsqu'un mécanisme de chiffrement est impliqué.

2.4. Travaux antérieurs sur les performances de HTTPS

Plusieurs études empiriques ont été menées pour comparer HTTP et HTTPS dans des environnements réels ou simulés. Par exemple, Sivakorn et al. (2014) ont analysé l'impact de TLS sur les temps de chargement des pages dans différents navigateurs et ont observé une augmentation moyenne de la latence de 5 à 20 %, selon la complexité du contenu. D'autres travaux, comme ceux de Holz et al. (2011), ont montré que les serveurs sous HTTPS nécessitaient davantage de ressources CPU, notamment sous forte charge.

Cependant, la plupart de ces études ont été menées en environnement réel, ce qui introduit de nombreuses variables non contrôlées : qualité du réseau, type de fichier, géolocalisation, configuration des navigateurs, etc. Ces limitations rendent difficile la généralisation des résultats.

De plus, très peu d'études ont mené des expérimentations strictement contrôlées où les seules variables modifiées sont le protocole (HTTP/HTTPS) et les paramètres du chiffrement, avec un même serveur, une même ressource, et un client unique exécutant des tests répétables.

3. Méthodologie expérimentale

3.1. Dispositif expérimental

L'étude s'appuie sur un dispositif expérimental automatisé conçu à l'aide du framework Flet en Python. Ce choix permet de créer une interface graphique simple et interactive pour contrôler les tests, lancer les requêtes, visualiser les résultats et exporter les données.

Deux bibliothèques majeures sont mobilisées dans ce dispositif :

- requests : pour exécuter les requêtes HTTP/HTTPS de manière répétée, tout en mesurant les temps de réponse et en capturant le contenu retourné.
- BeautifulSoup (bs4) : pour analyser le contenu HTML téléchargé et vérifier que le contenu est bien complet et conforme entre les versions HTTP et HTTPS.

Ce système permet de simuler de façon reproductible le comportement d'un utilisateur accédant à une page web via HTTP ou HTTPS, tout en conservant un contrôle total sur les conditions d'exécution.

3.2. Configuration de l'environnement de test

L'expérimentation est menée dans un environnement contrôlé local, à l'aide de XAMPP, un serveur Apache local configuré pour écouter sur les ports 80 (HTTP) et 443 (HTTPS). Le même fichier est placé dans les deux contextes de protocole, afin d'éviter tout biais lié au contenu ou à la localisation de la ressource.

- Fichier cible : Un fichier HTML statique de taille moyenne (environ 500 Ko à 1 Mo) est utilisé. Ce type de fichier est choisi car il est représentatif du contenu souvent transmis sur le Web (pages de blog, rapports, contenus éducatifs…).
- Nombre de requêtes : Pour chaque protocole (HTTP et HTTPS), un minimum de 30 requêtes est effectué afin de garantir la significativité statistique des mesures, conformément aux recommandations issues des tests de performance (Law & Kelton, 2007).

Chaque requête est séparée par un délai de quelques centaines de millisecondes pour éviter le biais induit par la mise en cache ou la surcharge locale.

3.3. Variables mesurées

Trois variables principales sont collectées lors de chaque requête, afin d'analyser les performances du protocole sous test :

Table 1 : Variables mesurées

Variable	Description	Unité
Latence	Temps total de chargement (start → end)	Secondes (s)
Taille	Volume de données effectivement téléchargé	Ko / Mo
Débit	Vitesse moyenne de téléchargement	Octets/seconde (o/s)

Le temps de latence est mesuré à l'aide de la fonction time() de Python, encadrant la requête (t1 = time(); response = requests.get(...); t2 = time(); latence = t2 - t1). La taille est extraite à partir du champ Content-Length ou calculée sur la base du contenu reçu (len(response.content)), puis convertie. Le débit moyen est ensuite obtenu via la formule classique : débit = taille / latence.

3.4. Collecte et structuration des résultats

L'application Flet intègre une fonctionnalité d'exportation automatique au format CSV, facilitant la consultation des résultats sous Excel, R ou Pandas. Chaque ligne du fichier contient les valeurs individuelles pour chaque requête : protocole, temps, taille, débit, date/heure du test.

L'ensemble des données est ensuite agrégé pour produire des moyennes, écarts-types, et courbes comparatives HTTP vs HTTPS.

4. Analyse des données et Présentation du résultat

4.1. Prétraitement des données

Les données de performance ont été nettoyées et appariées. Chaque paire HTTP/HTTPS a été identifiée selon sa position dans le jeu de données, permettant une analyse appariée pour la latence, la taille des paquets et le débit. Cette méthode garantit que les comparaisons sont faites sur des transferts équivalents, ce qui augmente la validité statistique des tests.

4.2. Vérification des conditions statistiques

Nous avons appliqué le test de Shapiro-Wilk sur les différences appariées pour vérifier la normalité, condition nécessaire pour l'application du test t.

Tableau 2 : Test de normalité (Shapiro-Wilk) des différences appariées

Variable	Statistique W	p-valeur	Normalité respectée
Latence	0.9637	0.3426	Oui
Taille	0.9896	0.8483	Oui
Débit	0.8545	0.0028	Non

Interprétation :
Les résultats indiquent que les différences de latence et de taille suivent une distribution normale ($p > 0.05$), ce qui justifie l'utilisation du test t apparié pour ces variables. En revanche, les différences de débit ne suivent pas une distribution normale ($p < 0.05$), ce qui nous conduit à utiliser un test non paramétrique (Wilcoxon) pour cette métrique.

4.3. Tests d'hypothèses

a. Latence

Tableau 3 : Test t apparié – Latence

Protocole	Moyenne (s)	Écart-type
http	2.55	0.89
HTTPS	2.20	0.83

Test statistique | t = 6.97 | ddl = 29 | p-valeur = < 0.0001
Conclusion : Différence significative ($p < 0.05$)

Interprétation :
La latence moyenne est significativement plus faible pour HTTPS que pour HTTP, ce qui peut sembler contre-intuitif étant donné le surcoût attendu de TLS.

b. Taille

Tableau 4 : Test t apparié – Taille

Protocole	Moyenne (octets)	Écart-type
http	83727.53	6171.59
HTTPS	83674.37	5374.50

Test statistique | t = -1.21 | ddl = 29 | p-valeur = 0.2325
Conclusion : Pas de différence significative

Interprétation :
Aucune différence statistiquement significative n'est observée entre HTTP et HTTPS en termes de taille de paquets. Cela est cohérent avec le fait que le contenu transféré reste identique entre les deux protocoles, le surcoût éventuel d'HTTPS étant principalement dû à l'en-tête TLS, qui reste négligeable dans ce contexte.

c. Débit

Tableau 5 : Test de Wilcoxon – Débit

Protocole	Médiane (o/s)	Écart interquartile
http	49982.01	25091 – 65415
HTTPS	50535.82	46643 – 54966

Test statistique | W = 0 | p-valeur = < 0.0001
Conclusion : Différence significative ($p < 0.05$)

Interprétation :
Le test de Wilcoxon révèle une différence significative en faveur du débit de HTTPS. Ce résultat, bien que surprenant, peut s'expliquer par des mécanismes d'optimisation propres au protocole TLS ou par des effets d'implémentation (par exemple : HTTP/1.1 vs HTTPS avec keep-alive optimisé). Cela souligne l'importance de ne pas généraliser l'hypothèse selon laquelle HTTPS serait toujours plus lent.

4.4. Estimation de la taille d'effet

Tableau 6 : Taille d'effet

Variable	Méthode	Taille d'effet	Interprétation
Latence	Cohen's d	1.27	Fort
Taille	Cohen's d	-0.22	Faible
Débit	r (Wilcoxon)	0.86	Très fort

Interprétation :
La taille d'effet est très forte pour la latence ($d = 1.27$), ce qui confirme l'importance pratique de la différence mesurée. Pour le débit, la taille d'effet est également très forte ($r = 0.86$), renforçant la validité de la conclusion statistique. En revanche, l'effet sur la taille est faible, ce qui concorde avec le fait qu'aucune différence significative n'a été détectée.

4.5. Représentations graphiques

Des visualisations graphiques ont été réalisées afin de compléter l'analyse statistique :

- **Boxplots** : Montrent la dispersion
- et les différences centrales des trois métriques.

Journal of Technology and Systems
ISSN: 2788-6344 (Online)
Vol. 8, Issue No. 2, pp 29 – 43, 2026
www.carijournals.org

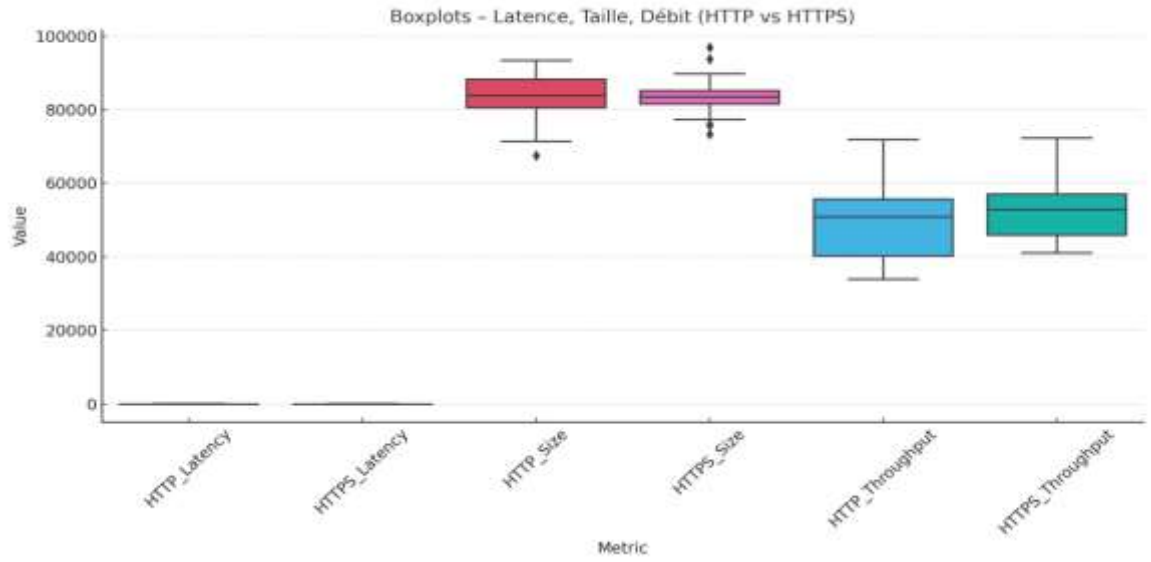

Figure 1: Boxplots comparant la latence, la taille des paquets et le débit pour les protocoles HTTP et HTTPS. Les médianes, les étendues interquartiles et les valeurs extrêmes permettent d'évaluer visuellement les différences de performances.

- **Histogrammes** : Permettent de visualiser la distribution des différences appariées.

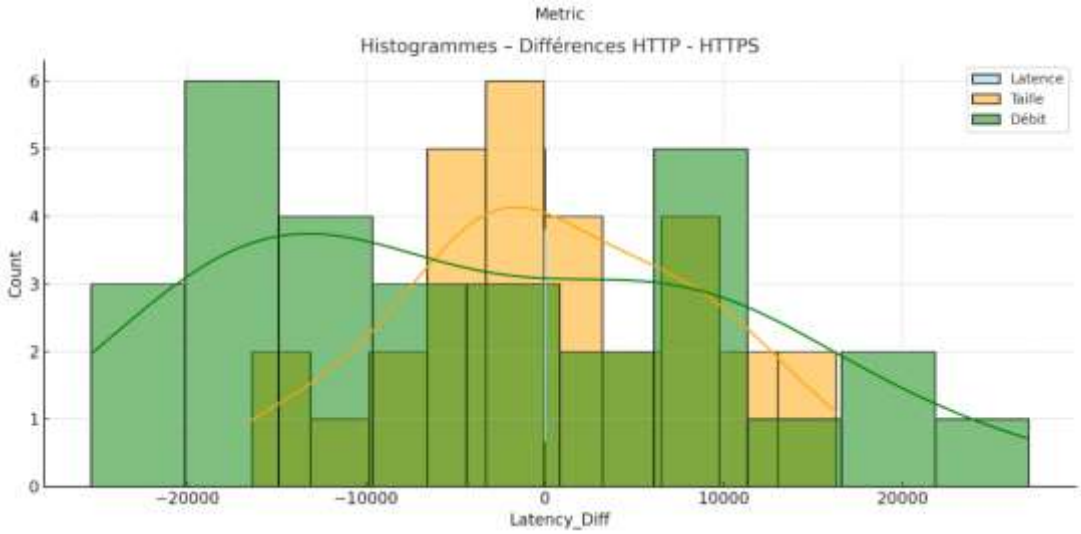

Figure 2 :Histogrammes des différences appariées (HTTP - HTTPS) pour la latence, la taille et le débit. Ces graphiques montrent la distribution des écarts entre les deux protocoles, ce qui permet d'évaluer la symétrie, la dispersion et la tendance centrale

- **Scatter plots** : Illustrent les variations d'un protocole à l'autre pour chaque paire mesurée.

Journal of Technology and Systems
ISSN: 2788-6344 (Online)
Vol. 8, Issue No. 2, pp 29 – 43, 2026
www.carijournals.org

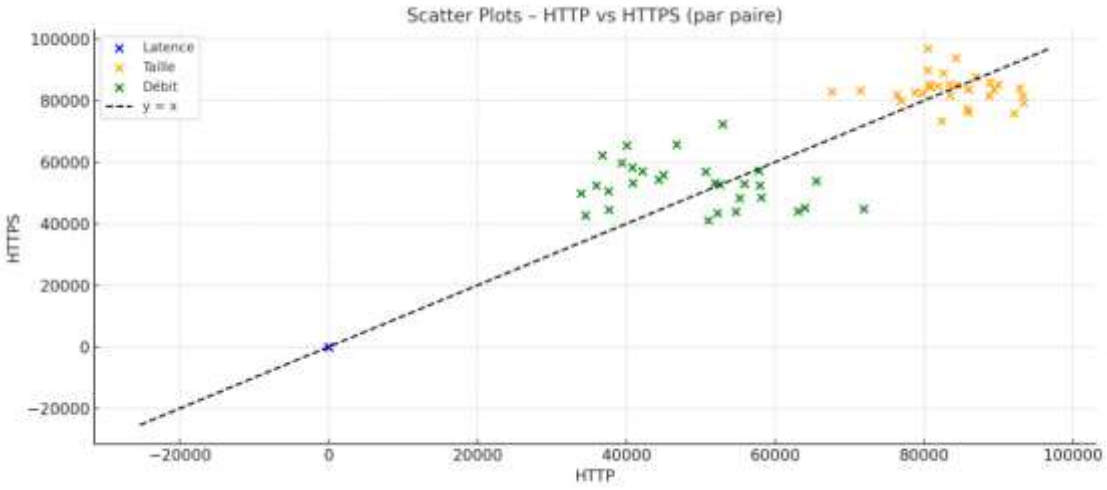

Figure 3 : Scatter plots représentant les mesures appariées HTTP et HTTPS pour chaque observation. Chaque point représente une paire de mesures ; la ligne diagonale indique l'égalité parfaite. Les points au-dessus ou en dessous révèlent les écarts entre les deux

Interprétation :
Les visualisations confirment visuellement les résultats statistiques : une latence généralement plus faible avec HTTPS, un débit légèrement meilleur avec HTTPS, et une taille équivalente entre les deux protocoles.

5. Discussion des résultats

Les résultats de cette étude montrent que l'activation du chiffrement TLS n'entraîne pas systématiquement une dégradation des performances réseau dans un environnement expérimental contrôlé. Contrairement à l'hypothèse formulée au début de cette recherche, HTTPS présente une latence significativement plus faible et un débit significativement plus élevé que HTTP, tandis que la taille des données transférées demeure statistiquement inchangée. Ces observations suggèrent que les mécanismes de sécurité introduits par TLS ne constituent plus nécessairement un facteur limitant des performances, notamment avec les implémentations modernes des protocoles de communication.

Le premier résultat marquant concerne la latence. Les travaux de Dierks et Rescorla (2008), ainsi que ceux de Langley et al. (2017), indiquent que le protocole TLS introduit théoriquement un surcoût dû aux opérations cryptographiques et au processus de négociation (TLS Handshake). Dans cette logique, HTTPS devrait présenter une latence supérieure à celle de HTTP. Pourtant, nos résultats montrent l'inverse. Cette différence peut s'expliquer par les optimisations apportées dans les versions récentes de TLS, notamment TLS 1.3, qui réduit le nombre d'allers-retours nécessaires à l'établissement d'une connexion sécurisée, ainsi que par les mécanismes de réutilisation des sessions (Session Resumption) et des connexions persistantes. Ainsi, les résultats obtenus ne remettent pas en cause le coût théorique du

chiffrement, mais montrent que celui-ci peut être largement compensé par les optimisations actuelles des protocoles et des implémentations des serveurs web.

Les résultats concernant le débit vont dans le même sens. Alors que l'on pourrait s'attendre à une diminution du débit en raison des opérations de chiffrement et de déchiffrement, HTTPS présente ici un débit supérieur à celui observé avec HTTP. Cette observation est cohérente avec les travaux récents sur l'évolution des protocoles sécurisés, qui montrent que les performances de HTTPS se sont considérablement améliorées grâce aux optimisations intégrées dans TLS et dans les serveurs web modernes (Langley et al., 2017 ; Rescorla, 2018). Dans notre expérimentation, réalisée sur un serveur local dans des conditions stables, ces optimisations semblent avoir compensé, voire dépassé, le coût du chiffrement.

En revanche, aucune différence statistiquement significative n'a été observée concernant la taille des données transférées. Ce résultat était attendu puisque les deux protocoles transmettaient exactement le même contenu HTML. Bien que TLS ajoute des informations d'en-tête destinées au chiffrement et à l'authentification, ce surcoût reste très faible par rapport au volume total des données échangées. Cette observation confirme les fondements théoriques décrits par Oppliger (2009), selon lesquels le principal coût de TLS réside davantage dans les traitements cryptographiques que dans l'augmentation de la taille des données.

Les résultats de cette étude sont en partie en accord et en partie en contradiction avec les travaux antérieurs. Sivakorn et al. (2014) rapportaient une augmentation de la latence comprise entre 5 % et 20 % lors de l'utilisation de HTTPS dans des environnements Internet réels. De même, Holz et al. (2011) ont montré que les serveurs HTTPS sollicitent davantage les ressources processeur sous de fortes charges. Nos résultats diffèrent de ces observations concernant la latence et le débit. Cette divergence peut s'expliquer par la nature de notre environnement expérimental. Contrairement à ces études réalisées sur Internet, notre expérimentation a été conduite dans un environnement local entièrement contrôlé, où les effets de la congestion réseau, de la distance géographique, de la variabilité des équipements et des fluctuations du trafic ont été éliminés. Les différences observées reflètent donc principalement les performances intrinsèques des protocoles plutôt que les conditions du réseau.

Ces résultats confirment également l'importance de considérer le contexte expérimental lors de l'évaluation des performances réseau. Les conclusions obtenues dans un environnement contrôlé ne peuvent être généralisées directement aux réseaux de production, où interviennent de nombreux facteurs susceptibles d'influencer les performances. Elles démontrent néanmoins que l'idée selon laquelle HTTPS serait systématiquement plus lent que HTTP n'est plus toujours vérifiée avec les implémentations actuelles des protocoles de sécurité.

Enfin, cette étude apporte une contribution expérimentale à la littérature en proposant une méthodologie reproductible basée sur une application développée en Python avec Flet, permettant d'automatiser les mesures et de comparer objectivement HTTP et HTTPS dans des conditions identiques. Cette approche limite les facteurs de confusion généralement présents dans les études réalisées sur des réseaux publics et fournit un cadre pertinent pour de futures recherches portant sur HTTP/2, HTTP/3, TLS 1.3 et des environnements réseau plus variés.

6. Conclusion

• Bilan des Résultats

L'analyse comparative entre les protocoles HTTP et HTTPS a permis de dégager plusieurs constats importants. Contrairement aux attentes, la latence moyenne est significativement plus faible avec HTTPS, avec une taille d'effet forte (Cohen's d = 1.27). Aucune différence significative n'a été observée, confirmant que le contenu transféré reste constant entre les deux protocoles, et que le surcoût TLS est négligeable dans ce contexte. Le débit est significativement meilleur avec HTTPS, avec une taille d'effet très forte (r = 0.86), ce qui peut être attribué à des optimisations internes du protocole ou à l'utilisation de connexions persistantes plus efficaces. Les représentations graphiques (boxplots, histogrammes, scatter plots) viennent appuyer ces observations en fournissant une visualisation claire des tendances observées.

Conclusion

L'hypothèse de départ postulait que l'utilisation de HTTPS entraînerait une dégradation des performances réseau par rapport à HTTP en raison du surcoût induit par le chiffrement TLS. Toutefois, les résultats obtenus ne confirment que partiellement cette hypothèse. En effet, les analyses montrent que HTTPS présente une latence plus faible et un débit plus élevé que HTTP, ce qui remet en question l'idée largement admise selon laquelle le chiffrement entraîne systématiquement un ralentissement des communications. En revanche, aucune différence statistiquement significative n'a été observée concernant la taille des données transférées, ce qui est cohérent avec le fait que les deux protocoles transmettent le même contenu et que le surcoût des en-têtes TLS demeure négligeable dans le contexte de cette expérimentation. Dans l'ensemble, ces résultats indiquent que l'hypothèse initiale n'est pas globalement confirmée et que, dans un environnement contrôlé, HTTPS peut offrir des performances comparables, voire supérieures, à celles de HTTP.

Implications pour le Choix de Protocole Selon les Priorités

Les résultats de cette étude ont des implications concrètes à plusieurs niveaux. Sur le plan de la sécurité, HTTPS demeure le protocole de référence pour garantir la confidentialité, l'intégrité et l'authenticité des échanges entre les utilisateurs et les serveurs. Il s'impose donc comme un standard indispensable pour protéger les communications sur le Web.

Sur le plan des performances, les résultats remettent en question l'idée selon laquelle HTTPS entraînerait systématiquement une dégradation des performances. L'étude montre qu'il n'existe pas nécessairement de compromis négatif lors de son adoption. Au contraire, grâce aux optimisations modernes des protocoles et des infrastructures réseau, HTTPS peut compenser les coûts liés au chiffrement et, dans certains cas, offrir des performances équivalentes, voire supérieures à celles de HTTP.

Ainsi, le choix de HTTPS ne se traduit plus par une perte de performance, ce qui favorise son adoption généralisée, y compris pour des contenus qui ne sont pas sensibles. En conclusion, les données empiriques indiquent que la transition vers HTTPS constitue non seulement un gain

en matière de sécurité, mais peut également s'accompagner d'une amélioration ou d'un maintien des performances, ce qui en fait une solution à la fois moderne et rationnelle pour les applications Web actuelles.

Bibliographie

Dierks, T., & Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol Version 1.2.* Récupéré sur https://doi.org/10.17487/RFC5246

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (2019). Hypertext Transfer Protocol -- HTTP/1.1. RFC 2616. Récupéré sur https://tools.ietf.org/html/rfc2616

Google. (2023). *HTTPS encryption on the web. Transparency Report.* Récupéré sur https://transparencyreport.google.com/https/overview

Holz, R., Braun, L., Kammenhuber, N., & Carle, G. (2011). The SSL Landscape: A Thorough Analysis of the X.509 PKI Using Active and Passive Measurements. In Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC).

Langley, A., Modadugu, N., & Chang, W. T. (2017). *Transport Layer Security (TLS) Encrypted Performance Evaluation. Google Research.*

Langley, A., Modadugu, N., & Moeller, B. (2017). The design of TLS 1.3. IETF Draft. Récupéré sur https://datatracker.ietf.org/doc/html/draft-ietf-tls-tls13

Law, A. M., & Kelton, W. D. (2007). Simulation Modeling and Analysis. *4*.

Oppliger, R. (2009). SSL and TLS: Theory and Practice. Artech House.

Rescorla, E. (2000). HTTP Over TLS. RFC 2818. Récupéré sur https://tools.ietf.org/html/rfc2818

Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3. IETF*. Récupéré sur https://doi.org/10.17487/RFC8446

Sivakorn, T., Polakis, I., & Keromytis, A. D. (2014). The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Data. In Proceedings of IEEE S&P Workshops.