

UDC 004.738.5:004.451

DOI: <https://doi.org/10.17721/3041-2323.2024.449-455>

Vadym SVARICH, Student

ORCID ID: 0009-0000-3816-2804

e-mail: svarich.vadim@gmail.com

Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

## COMPREHENSIVE ANALYSIS OF TCP/IP, UDP, AND HTTP COMMUNICATION PROTOCOLS IN MODERN NETWORKING

*This paper presents a comprehensive analysis of the communication protocols TCP/IP, UDP, and HTTP within the context of modern networking. The study examines their fundamental principles, strengths, weaknesses, and recent advancements. Performance characteristics such as latency, throughput, packet loss resilience, protocol overhead, and scalability are evaluated through practical testing and analytical methods. The results provide insights for optimizing protocol selection and usage in various contemporary network applications, including IoT, streaming services, and cloud computing environments.*

**Keywords:** *communication protocols, TCP/IP, UDP, HTTP, network performance, protocol analysis, modern networking, IoT, cloud computing.*

### **Background**

The rapid advancement of information technology – including the proliferation of the Internet of Things (IoT), cloud computing, and high-speed networking – has significantly increased the importance of effective and reliable communication protocols (Forouzan, 2017; Kurose, & Ross, 2017). TCP/IP, UDP, and HTTP are foundational protocols that underpin data transmission across networks of various scales and purposes (Comer, 2018). Understanding their characteristics, functionality, and recent enhancements is critical for optimizing network applications and improving overall communication system performance in the context of current technological trends.

The objective of this research is to conduct a comprehensive and analytical analysis of the TCP/IP, UDP, and HTTP protocols, including their fundamental principles, advantages, disadvantages, and recent developments. The study includes performance evaluation based on latency, throughput, packet loss resilience, protocol overhead, and

© Svarich Vadym, 2024

scalability through practical testing and analytical assessment. The research aims to provide recommendations for protocol usage in modern network applications.

***Theoretical Basis of Communication Protocols. Overview of TCP/IP, UDP, and HTTP.*** TCP/IP (Transmission Control Protocol/Internet Protocol) is a suite of protocols that provides reliable, connection-oriented data transmission over the Internet. TCP ensures reliable communication by handling connection establishment, flow control, congestion control, error detection, and retransmission of lost packets (Forouzan, 2017). Recent enhancements to TCP include algorithms for improved congestion control, such as BBR (Bottleneck Bandwidth and Round-trip propagation time), which optimizes performance over high-speed networks (Cardwell et al., 2017).

UDP (User Datagram Protocol) is a lightweight, connectionless protocol that enables fast data transmission without the overhead of error checking and flow control, making it suitable for applications where speed is more important than reliability (Kurose, & Ross, 2017). Modern applications increasingly use UDP in protocols like QUIC, which aim to combine UDP's speed with mechanisms for reliability and security at the application layer (Iyengar, & Thomson, 2021).

HTTP (HyperText Transfer Protocol) is an application-level protocol used for transmitting hypertext documents over the Internet. Traditionally operating over TCP, HTTP ensures reliable web content transmission (Comer, 2018). The recent development of HTTP/2 and HTTP/3 has introduced significant enhancements, including multiplexing, header compression, and, in the case of HTTP/3, operating over QUIC (UDP-based) to improve performance and reduce latency in web communications (Fielding, & Reschke, 2014; Thomson, & Bishop, 2022).

***Research Methodology. Description of the Conducted Tests.*** Testing communication protocols involves evaluating their performance based on various metrics to understand their efficiency, reliability, and scalability in modern networking environments. This section describes the different types of tests performed and how they are implemented using code in `sender.js`, along with the analytical methods employed.

***Latency Comparison Tests.*** Objective: Measure the time it takes for data to travel from the sender to the receiver and back (Round-Trip Time or RTT) under different network conditions and packet sizes.

Test Description:

- For each protocol (TCP, UDP, HTTP), data packets of varying sizes (e. g., 64 bytes to 1 MB) are sent multiple times under controlled network conditions.
- Network conditions such as bandwidth, delay, jitter, and packet loss are emulated using network simulation tools (e.g., NetEm in Linux) to mimic real-world scenarios.
- High-precision timers record the time taken for data to reach the receiver and return an acknowledgment (if applicable).
- Testing functions (e. g., "testTCP", "testUDP", "testHTTP") handle data transmission and latency measurement, ensuring synchronization and accurate timekeeping.
- The average latency for each protocol and data size is calculated and statistically analyzed, considering standard deviation and confidence intervals.

$$Latency = \frac{\sum_{i=1}^N Latency_i}{N}.$$

**Throughput Comparison Tests.** Objective: Measure the amount of data successfully transmitted over the network per unit of time, evaluating the protocols' efficiency under varying network loads and conditions.

Test Description:

- Data streams of significant duration are sent through TCP, UDP, and HTTP protocols.
- The amount of data received during a set time period is measured, considering any retransmissions or packet losses.
- Throughput is calculated based on the received data and time taken, adjusted for protocol overhead.
- Test functions handle data transmission, throughput measurement, and error handling.
- Analytical methods assess throughput performance, including plotting throughput over time and under different network conditions.

$$Throughput = \frac{DataSize}{Time}.$$

Additional Tests:

- Packet Loss Resilience Tests: Evaluate how each protocol handles packet loss by introducing controlled packet loss rates and measuring the impact on data transmission.

- **Protocol Overhead Analysis:** Calculate the overhead introduced by each protocol in terms of additional bytes transmitted (headers, control messages) and processing time.
- **Scalability Tests:** Assess how each protocol performs as the number of connections or data flows increases, particularly relevant in high-traffic or large-scale applications.

### **Results**

The study revealed different performance characteristics for each protocol under various network conditions.

#### **Comparison of Data Transmission Latencies**

**TCP** Average latency: 5.10 ms, Minimum latency: 2.00 ms, Maximum latency: 12.00 ms.

**UDP** Average latency: 1.28 ms, Minimum latency: 0.50 ms, Maximum latency: 3.00 ms.

**HTTP** Average latency: 0.77 ms, Minimum latency: 0.30 ms, Maximum latency: 2.00 ms.

The latency values are averages across multiple tests with varying packet sizes and network conditions. Statistical analysis includes standard deviations and confidence intervals.

#### **Comparison of Network Throughput**

**TCP** Average throughput: 85 Mbps, Stability: High in stable networks, decreases in high-latency or lossy networks.

**UDP** Average throughput: 95 Mbps., Stability: Consistent but may result in higher data loss in unreliable networks,

**HTTP** Average throughput: 80 Mbps (HTTP/1.1), 90 Mbps (HTTP/2), 95 Mbps (HTTP/3), Stability Improved with HTTP/2 and HTTP/3 due to optimizations like multiplexing and reduced overhead.

#### **Analysis of Data Transmission Latency**

TCP has higher latency due to connection establishment (three-way handshake), congestion control, and error recovery mechanisms. In high-latency networks, these mechanisms can introduce significant delays.

UDP offers lower latency by avoiding connection establishment and control mechanisms, making it suitable for time-sensitive applications such as online gaming and VoIP.

HTTP, particularly in its modern versions (HTTP/2 and HTTP/3), shows improved latency due to optimizations like multiplexing, header compression, and, in the case of HTTP/3, operating over QUIC (UDP-

based). This reduces latency by eliminating TCP's handshake and leveraging UDP's speed while implementing reliability at the application layer.

#### **Protocol Throughput**

TCP can achieve high throughput in stable networks due to its congestion control algorithms but may suffer in lossy networks where retransmissions are frequent.

UDP maintains consistent throughput but at the risk of data integrity, as it lacks mechanisms for error correction and retransmission.

HTTP/2 and HTTP/3 improve throughput over HTTP/1.1 by enabling multiple concurrent streams over a single connection and reducing overhead.

#### **Packet Loss Resilience**

TCP is superior due to its retransmission mechanisms and acknowledgment of received packets, ensuring data integrity.

UDP lacks inherent packet recovery, leading to potential data loss in unreliable networks unless handled by the application layer.

HTTP/3 (QUIC) incorporates mechanisms to handle packet loss more efficiently than TCP, reducing the need for retransmissions and improving performance.

#### **Protocol Overhead**

TCP introduces overhead due to its extensive control mechanisms, which can impact performance in high-speed networks with small packet sizes.

UDP has minimal overhead, making it efficient for applications that transmit small amounts of data frequently.

HTTP/2 and HTTP/3 reduce overhead through header compression (HPACK and QPACK) and eliminate redundant information transmission.

#### **Modern Aspects**

QUIC Protocol. Developed by Google and standardized by the IETF as HTTP/3, QUIC combines UDP's speed with reliability and security mechanisms at the application layer. It reduces connection establishment time and improves performance over high-latency networks.

TCP Congestion Control Algorithms. New algorithms like BBR optimize performance over high-speed, long-distance networks by modeling the network's bandwidth and latency, leading to better utilization of available capacity.

Edge Computing and IoT. The scalability and efficiency of protocols are increasingly important in IoT and edge computing environments, where resources are constrained, and networks are variable.

### **Discussion and conclusions**

This study provides a comprehensive analysis of TCP/IP, UDP, and HTTP protocols, highlighting their performance characteristics under various network conditions. TCP offers reliable, ordered, and error-checked delivery of data but incurs higher overhead and latency due to its connection-oriented nature and control mechanisms (Forouzan, 2017). UDP is fast and efficient, suitable for applications where speed and low latency are essential, and occasional data loss is acceptable (Kurose, & Ross, 2017). HTTP, especially in its modern iterations (HTTP/2 and HTTP/3), incorporates optimizations that enhance performance for web applications, reducing latency and improving throughput (Fielding, & Reschke, 2014; Thomson, & Bishop, 2022).

The choice of protocol has significant implications for network performance, particularly in modern applications such as streaming, real-time communications, and large-scale distributed systems. TCP is recommended for critical data transmission where reliability, order, and data integrity are paramount, such as file transfers, emails, and database transactions. UDP is suitable for real-time applications where speed and low latency are essential, such as live video streaming, online gaming, and VoIP. HTTP/3 should be considered for web services and applications requiring both reliability and reduced latency, benefiting from QUIC's features over UDP (Iyengar, & Thomson, 2021; Thomson, & Bishop, 2022). In modern network designs, evaluating the use of emerging protocols like QUIC can lead to performance improvements by combining the benefits of UDP's speed with mechanisms for reliability and security.

Future work includes investigating the performance of newer protocols like QUIC in comparison to traditional TCP and UDP under various network conditions, analyzing the impact of different TCP congestion control algorithms on network performance, especially in high-speed networks, and exploring the use of these protocols in IoT and edge computing environments where resource constraints and network variability are significant.

### **References**

Cardwell, N., Cheng, Y., Gunn, C. S. et al. (2017). BBR: Congestion-based congestion control. *Communications of the ACM*, 60(2), 58–66.

Comer, D. E. (2018). *Computer networks and internets* (6th ed.). Pearson Education.

Fielding, R. T., & Reschke, J. (2014). *Hypertext Transfer Protocol (HTTP/1.1): Message syntax and routing* (RFC 7230). IETF.

Forouzan, B. A. (2017). *Data communications and networking* (5th ed.). McGraw-Hill Education.

Iyengar, J., & Thomson, M. (2021). *QUIC: A UDP-based multiplexed and secure transport* (RFC 9000). IETF.

Kurose, J. F., & Ross, K. W. (2017). *Computer networking: A top-down approach* (7th ed.). Pearson.

Thomson, M., & Bishop, M. (2022). *Hypertext Transfer Protocol Version 3 (HTTP/3)* (RFC 9114). IETF.

**Отримано редакцією журналу / Received: 16.09.24**

**Прорецензовано / Revised: 26.09.24**

**Схвалено до друку / Accepted: 01.10.24**

**Вадим СВАРИЧ, студ.**

**ORCID ID: 0009-0000-3816-2804**

**e-mail: svarich.vadim@gmail.com**

**Київський національний університет  
імені Тараса Шевченка, Київ, Україна**

## **КОМПЛЕКСНИЙ АНАЛІЗ КОМУНІКАЦІЙНИХ ПРОТОКОЛІВ TCP/IP, UDP ТА HTTP У СУЧАСНИХ МЕРЕЖАХ**

*Представлено комплексний аналіз комунікаційних протоколів TCP/IP, UDP та HTTP у контексті сучасних мережеских технологій. У дослідженні розглянуто їхні основні принципи, переваги, недоліки й останні досягнення. Оцінено характеристики продуктивності, такі як затримка, пропускна здатність, стійкість до втрати пакетів, накладні витрати протоколу та масштабованість, із використанням практичних тестів та аналітичних методів. Отримані результати дають змогу оптимізувати вибір протоколу та його використання у різних сучасних мережеских застосуваннях, зокрема і в Інтернеті речей (IoT), потокових сервісах і хмарових обчисленнях.*

**Ключові слова:** *комунікаційні протоколи, TCP/IP, UDP, HTTP, продуктивність мережі, аналіз протоколів, сучасні мережі, Інтернет речей (IoT), хмарові обчислення.*

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.