

A Comparative Study of TCP and UDP Performance Using NS-3 Simulation

Kajian Perbandingan Prestasi TCP dan UDP Menggunakan Simulasi NS-3

*Guo Gaoyang, Cui Zhiyi, Faizan Qamar, Azana Hafizah Mohd Aman**

Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), 43600 UKM Bangi, Selangor, Malaysia.

**Corresponding author: Azana Hafizah Mohd Aman (azana@ukm.edu.my)*

Received 22 June 2024

Accepted 24 October 2025, Available online 30 June 2025

ABSTRACT

As the center of the protocol layer, the transport layer is independent of the current network and transports data from the source device to the target device reliably and efficiently. Transmission Control Protocol (TCP) is a component of the transport layer that uses Internet Protocol (IP) protocol to achieve end-to-end data transmission. This paper investigates the experimental evaluation of TCP performance in recent years. Most of the studies get the performance results of TCP by comparing it with User Datagram Protocol (UDP). Therefore, in the simulation test conducted in this paper, the performance of TCP is judged by the performance comparison method of TCP and UDP. In this simulation, the NS-3 simulator was selected to complete the simulation. Choose throughput, end-to-end delay, lost packets, packet loss ratio, bandwidth utilization, jitter, and packet delivery ratio to perform numerical calculations. Then compare and summarize these performance metrics. Experimental results show that TCP has good performance, but its performance is affected by bandwidth, network delay, and other factors. Finally, the experimental results are organized and discussed, then plan the future work.

Keywords: NS-3; TCP; UDP; Network Parameters; Network Performance

ABSTRAK

Sebagai pusat lapisan protokol, lapisan pengangkutan adalah bebas daripada rangkaian semasa dan mengangkut data dari peranti sumber ke peranti sasaran dengan pasti dan cekap. Transmission Control Protocol (TCP) ialah komponen lapisan pengangkutan yang menggunakan protokol Internet Protocol (IP) untuk mencapai penghantaran data hujung ke hujung. Kertas kerja ini menyiasat penilaian eksperimen prestasi TCP dalam beberapa tahun kebelakangan ini. Kebanyakan kajian mendapat keputusan prestasi TCP dengan membandingkannya dengan User Datagram Protocol (UDP). Oleh itu, dalam ujian simulasi yang dijalankan dalam kertas kerja ini, prestasi TCP dinilai dengan kaedah perbandingan prestasi TCP dan UDP. Dalam ujian ini, simulator NS-3 telah dipilih untuk melengkapkan simulasi. Pilih throughput, kelewatan hujung ke hujung, paket hilang, nisbah kehilangan paket, penggunaan lebar jalur, jitter dan nisbah penghantaran paket untuk melakukan pengiraan

berangka. Kemudian bandingkan dan ringkaskan metrik prestasi ini. Keputusan percubaan menunjukkan bahawa TCP mempunyai prestasi yang baik, tetapi prestasinya dipengaruhi oleh lebar jalur, kelewatan rangkaian dan faktor lain. Akhirnya, keputusan eksperimen disusun dan dibincangkan, kemudian merancang kerja masa depan.

Kata Kunci: NS-3; TCP; UDP; Parameter Rangkaian; Prestasi Rangkaian

INTRODUCTION

Network protocol refers to a series of rules and standards which is used to regulate the process of data exchange and communication in computer networks (Bertsekas & Gallager, 2021). In the network, devices from different vendors and platforms need to communicate with each other. Network protocols define a unified standard for data exchange so various devices can be closely integrated into the same network environment. At the same time, the protocol provides error detection and correction mechanisms to ensure that the data can be transmitted in the correct order and integrity. With the development of the network, data security risks are rising rapidly. Network protocols use mechanisms such as encryption and authentication to protect data from unauthorized access (Arogundade, 2023). On the Internet, protocols are usually related to the layer to which a process belongs in the network model. This paper uses TCP/IP four-layer network model to show the protocols of different layers. And correspondence between different layers and protocols is in Figure 1.

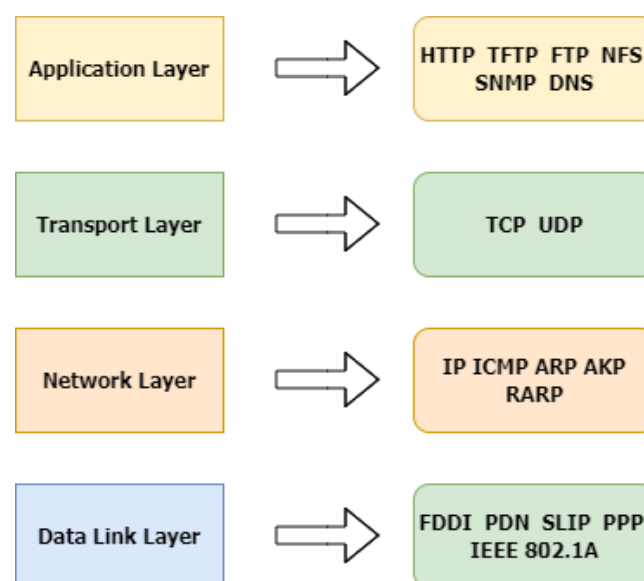


FIGURE 1. Protocols corresponding to each layer of the TCP/IP Model

TCP is one of the most important and basic protocols on the Internet. It is a connection-oriented, byte stream-based transport layer communication protocol responsible for reliably transporting data over a network (Postel, 1981). During data transmission, the two communicating parties start transmitting data after establishing the connection. When congestion and data loss occur in the network, TCP manages the data transmission rate through flow control and congestion control mechanisms. In the data transmission, the data is verified to ensure the integrity of the data. When the data error is detected, the data will be retransmitted to ensure the accuracy of data transmission. These mechanisms also make TCP very reliable and hence the preferred transport protocol for a variety of use cases. In real-world applications, TCP is often used to acknowledge data transfer, such as Web page loading and data transfer, and many instant

messaging applications. Email protocols such as Simple Mail Transfer Protocol (SMTP) and Post Office Protocol (POP3) use TCP to transmit and receive mail (Murkomen, 2024). TCP is also involved in remote connection work to provide users with secure remote login capabilities.

UDP is a packet-oriented transport layer protocol, which is easier and more flexible than TCP to transfer data. It provides a process for applications to send messages to other programs with minimal protocol mechanism but does not guarantee transmission and duplication protection (Postel, 1980). UDP is a connectionless protocol, that can send data packets directly without establishing a connection before communication, so it is widely used in applications with real-time requirements. Although UDP transfers data quickly, it does not perform error checking when sending data, nor retransmit in case of transmission errors. As a result, data may be lost or out of order (Lee et al., 2017).

NS-3 is a simulator that can simulate various types and sizes of real-world networks on a single computer (Campanile et al., 2020). It provides a wide range of network models and protocol implementations, including various transport layer protocols such as TCP and UDP. At the same time, it supports users to customize the network model and protocol implementation according to their needs. The basic architecture of NS-3 is in Figure 2.

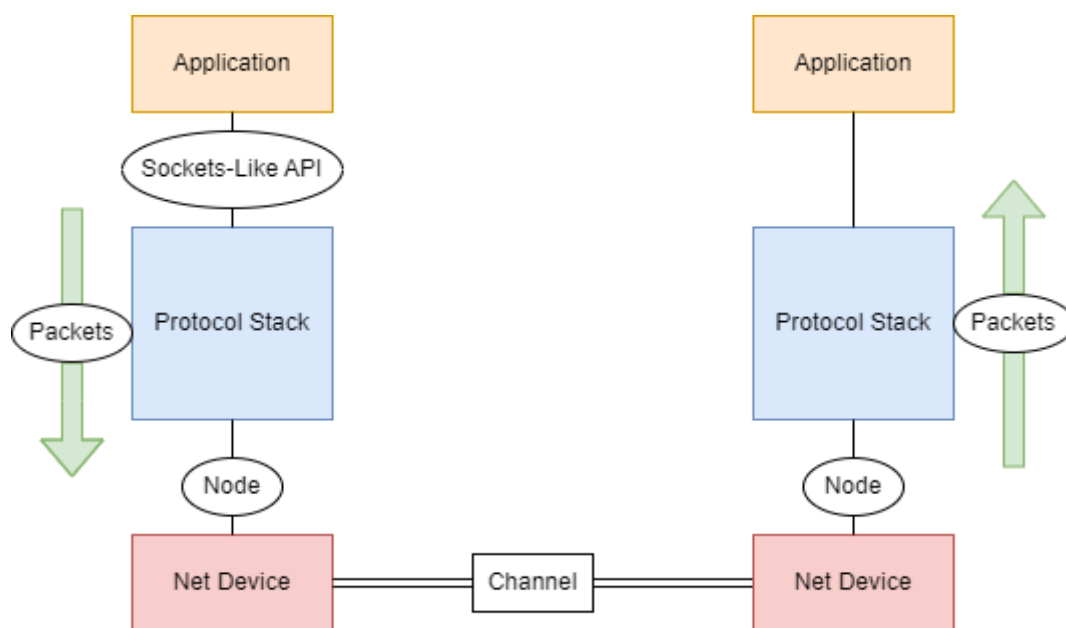


FIGURE 2. NS-3 basic model

TCP congestion control tracks a variable named congestion window (CWND), which limits the number of data TCP can send to the network before it receives the Acknowledge character (ACK), and the receive window (RWND), which tells the receiver how much data it can accept. When $CWND > RWND$, the ability of the receiver to limit the send window reaches its maximum. When $CWND < RWND$, the network congestion limit sending window reaches the maximum value. TCP congestion control algorithms include slow-start, congestion avoidance, fast recovery, and fast retransmission (Allman et al., 2009).

1. Slow-start:

TCP has a slow-start process immediately after establishing the connection. The CWND value is initialized to a maximum segment size (MSS), and the CWND value is incremented to

two MSS each time a packet is transmitted. After the two segments are successfully transmitted, each segment is increased by one, so there are four MSS. And so on, the value of CWND is doubled on each success until a threshold is reached (the congestion avoidance threshold) (Stevens, 1997). The congestion avoidance threshold (CAT) is an important parameter in the TCP congestion control algorithm. In TCP communication, when there is congestion in the network, TCP will take a series of measures to alleviate the congestion and maintain the network performance. The congestion avoidance threshold is the threshold used to determine the transition of TCP from the slow-start state to the congestion avoidance state. A diagram of TCP slow-start is in Figure 3.

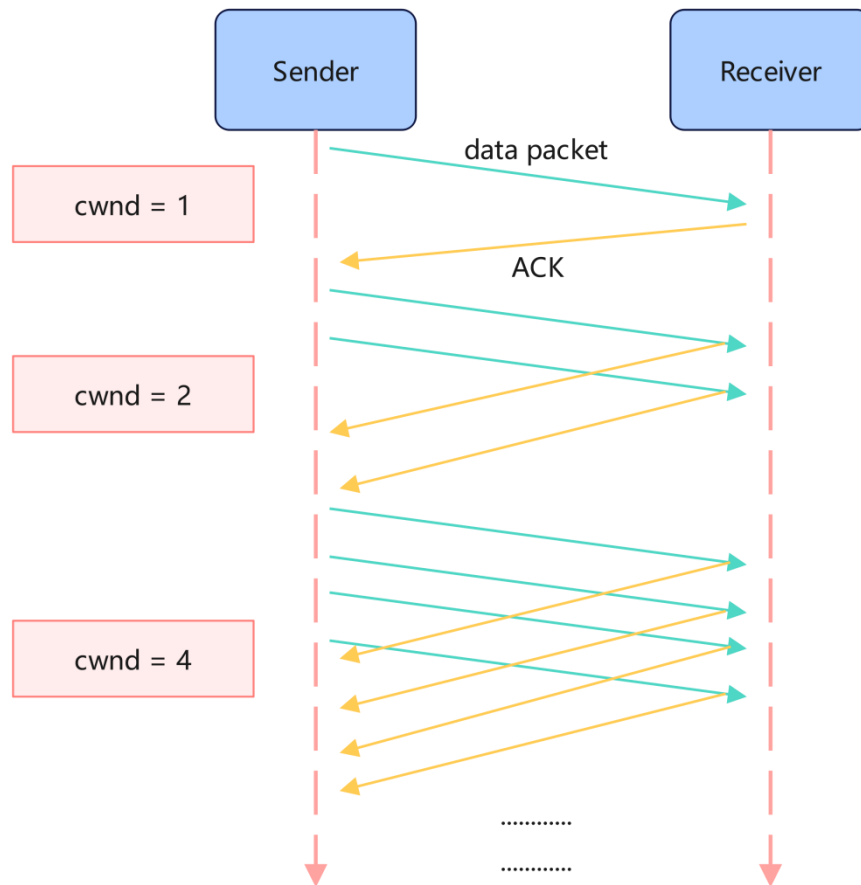


FIGURE 3. Diagram of TCP slow-start

2. Congestion Avoidance:

Slow start threshold (ssthresh) is used to determine the threshold at which the phase of TCP transitions from the slow start to congestion avoidance (Allman et al., 2009). When $CWND < ssthresh$, TCP uses the slow-start algorithm. When $CWND \geq ssthresh$, TCP enters the congestion avoidance phase. The value of CWND cannot be doubled each time a packet arrives, so TCP goes into the congestion control state.

3. Fast Recovery:

After the CWND value is halved by TCP, it still receives three redundant acks, $ssthresh = CWND/2$, so $CWND = ssthresh + 3$. For the losing packet segments that bring TCP into the fast recovery state, the value of CWND is increased by 1 MSS for each received redundant ACK. When 1 ACK for the lost packet segment arrives, TCP enters the congestion avoidance state after lowering CWND. If a timeout occurs after the congestion control state,

then a transition to the slow-start state is made, the value of CWND is set to 1 MSS, and the value of ssthresh is set to half that of CWND (Haile et al., 2021).

4. Fast Retransmission:

When the sender receives three duplicate acks in a row, it will assume that a packet segment is lost and immediately retransmit the lost packet segment without having to wait for a timeout (Abadleh et al., 2022). A diagram of TCP fast retransmission is in Figure 4.

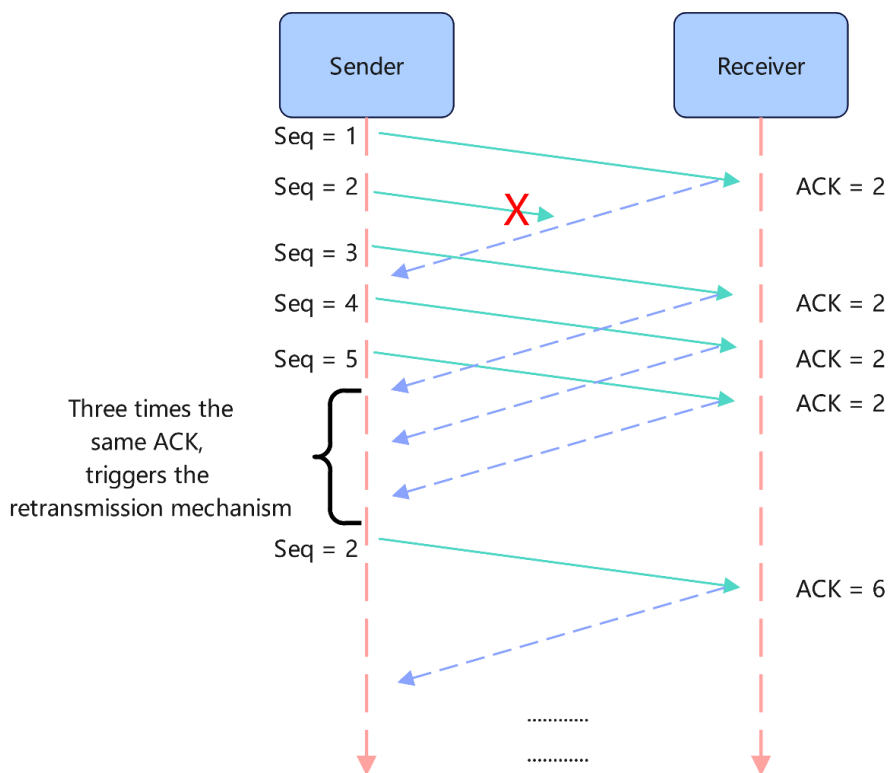


FIGURE 4. Diagram of TCP fast retransmission

CONTRIBUTION

This paper surveys the research during these years and finds a mass of new protocols and environments using NS-3 to compare the performance of TCP and UDP, to prove whether it is better than previous protocols or environments. The survey also found some limitations in these studies. For example, the NS-2 simulator has limitations in simulating the complexity and dynamics of the network, protocol simulation will be affected by factors such as simulation parameters, and incomplete parameters cannot fully understand the performance of TCP and UDP. On this basis, the following are some specific contributions of this paper:

1. Designed an NS-3 simulation environment to compare the performance of TCP and UDP in this article.
2. The experiment observed the performance of the two protocols by establishing end-to-end network topology and setting different transmission data to it.
3. Throughput, end-to-end delay, lost packets, packet loss ratio, bandwidth utilization, jitter, and packet delivery ratio are used as performance metrics, and it was found that TCP outperforms UDP under these seven detection indicators.

The remainder of this paper is as follows. The third section investigates the performance of

TCP in recent years. The fourth section introduces the experimental process and experimental results in detail. The fifth section discusses the simulation results. The sixth section summarizes the paper. The seventh section outlines future work.

RELATED WORKS

Many of the performance results for TCP have been measured in parallel simulation studies with other protocols, such as UDP. Therefore, a general understanding of the performance of TCP can be obtained by comparing several experiments. The experiment shows that TCP generally takes more time than UDP to transfer data. This is because a three-way handshake between the source and destination addresses is performed before data can be sent. When TCP sends a data segment, it sets a timer and then waits for the receiver to accept the packet. If the sender does not receive the feedback from the receiver within the time set by the timer, it retransmits the segment (Yaun et al., 2003).

The study by (Simon et al., 2023) shows the simulation experiment of TCP and UDP in the NS-3 simulator environment. The authors divided the experiments into two groups, the first with a fixed packet size of 64 bytes and varying bandwidth from 0.1 Mb/ms to 0.5 Mb/ms. The second group has a fixed bandwidth size of 0.3 Mb/ms. The size of the data packet is varied and the range is active between 800 and 1000 bytes. The experimental setup was simulated in an environment with 8 nodes and 60 seconds. During the simulation, the authors choose two performances, delay, and throughput, as the basis for comparison. The resulting graph shows that the latency of TCP is around 0.829, while the latency of UDP is as high as around 2.025. The throughput of TCP fluctuates around 9.737, while that of UDP fluctuates around 4.854. Thus, the delay of TCP is better than that of UDP, while the throughput is higher.

The authors (AL-Dhief et al., 2018) analyze the performance of TCP and UDP using NS-2 in two scenarios. Two sets of experiments were carried out with the same number of nodes and simulation time. The first set is carried out under different bandwidth environments. The experimental data show through comparison, that the throughput difference of TCP is 13.61, the end-to-end delay difference is -0.36358, and the packet loss ratio of TCP is 0%. So it is clear that TCP has better performance. Different sizes of packets are used in the second experiment. The data show that the throughput difference between TCP and UDP is 278. End-to-end delay difference -1.90719. TCP achieves a packet loss value of 4.74, while UDP achieves a packet loss value as high as 97. From the two sets of experiments, TCP performs much better than UDP.

In a study (Soomro et al.), the authors assess TCP and UDP under Energy Priority and Link Aware Ad-hoc On-demand Distance Vector (EPLAODV) routing protocol. In the simulation experiment, NS-2 is selected as the main parameter according to network traffic load, node mobility, and simulation time. The experiment evaluates the performance of PDR, energy consumption, and end-to-end delay. As the network load traffic increases, more activities establish connections and the network becomes more congested. In this case, the packet loss ratio will increase greatly, while the data show that the TCP packet transmission rate remains stable, and the UDP packet transmission rate will decrease. The delay of TCP traffic is smaller than UDP traffic. In the aspect of energy consumption, both TCP and UDP tend to be stable, but TCP consumes less energy. When the simulation time increases, there is no packet loss ratio due to the packet retransmission mechanism provided by TCP, while the packet transmission rate of UDP decreases by 80%. At the same time, when the mobility of nodes increases, more data packets will be dropped due to factors such as link failures, so UDP will also consume more node energy.

In many practical applications, comparing the performance of TCP and UDP has become a common way to test the efficiency of protocols. In the literature (Segara & Ramadhani, 2023), the experiment proceeded in the environment of warship communication, and the data packet transmission rate and delay of TCP and UDP were compared to determine whether the Ad hoc On-Demand Distance Vector (AODV) protocol was reliable and efficient. The NS simulator was chosen for the experiment. The number of nodes was set to 10 and 21 nodes, respectively, and the proportion was 1000 m x 1000 m for 200 seconds. The experiments were divided into two groups based on scene conditions. The first set of simulation results show that the delay of UDP is higher than TCP. In terms of packet transmission rate, the result value of UDP tends to decrease, while the result of TCP tends to have a stable value after the decrease. The latency of TCP and UDP in the second group shows a decreasing trend. The packet transmission rate will be affected by the type of traffic and width of the region. Data show that with the rise in the number of nodes, the PDR of TCP will gradually stabilize at a fixed value from the beginning. However, TCP has a higher packet delivery ratio than UDP, so it is more suitable for reliable and success-first communication.

In (Kumar & Verma, 2022), based on the existing Optimized Link State Routing (OLSR) scheme, the authors propose a new routing scheme, Airborne Optimized Link State Routing (AOLSR), which can reduce the routing overhead. The new scheme is compared with the OLSR scheme by simulating the performance of TCP and UDP with NS-3. The experimental setup time is 20s, the packet size is 512 bytes, the transmission packet interval is 0.008 seconds, and 5000 packets are transmitted in the experiment. The experimental results show that the packet delivery rate of TCP in the AOLSR environment maintains about 85%, and the packet delivery rate of UDP declines from 80%. With the increase of node speed, the packet delivery ratio of both of them declines and finally tends to a stable value. The delay of both is smooth at first and then increases with the increase of node speed. The latency is within 0.002 milliseconds for TCP and 0.003 milliseconds for UDP. When the node speed is greater than 300m/s, the routing overhead of UDP is more than twice that of TCP, but the throughput of UDP is higher than TCP.

The packet delivery ratio of TCP is 20% higher in the OLSR environment. As the node speed grows, the end-to-end delay of TCP remains around 0.002 milliseconds. UDP, on the other hand, starts to increase in latency at 0.003 milliseconds, and so does its routing overhead, while TCP remains stable. The packet delivery ratio of TCP is better than that of UDP in these two protocols. The performance of TCP is significantly better than UDP. A summary of each literature is shown in Table 1.

TABLE 1. Summary of the TCP and UDP literature survey

Ref.	Method and Parameter	Result	Advantage	Limitation
(Simon et al., 2023)	- NS-2 simulator - Throughput and end-to-end delay	Latency: TCP is around 0.829 UDP is around 2.025 Throughput: TCP is around 9.737 UDP is around 4.854	Changing two network parameters, bandwidth and packet size, helps to fully evaluate the performance of the two protocols.	The range of parameters and the number of nodes are fixed, which cannot simulate all the network environments and the operation performance of large-scale networks.

(AL-Dhief et al., 2018)	<ul style="list-style-type: none"> - NS-3 simulator - Throughput, end-to-end delay, packet delivery rate, and packet loss 	<p>Throughput: TCP is 700.7 UDP is 687.1</p> <p>End-to-end latency: TCP is 0.62018 seconds UDP is 0.98376 seconds</p> <p>Packet delivery rate and packet loss ratio: PDR: TCP is 100% network utilization, and UDP is 4.04.</p> <p>Packet loss: TCP is 0%, UDP is 95.96%.</p>	<p>Performance evaluation using a variety of metrics compares the performance differences between TCP and UDP. It helps to understand and select protocols that are suitable for specific application scenarios.</p>	<p>NS-2 does not fully simulate all the complexity and dynamics of a real network environment. In practical use need to consider the diversity of the protocol, implementation cost, and security.</p>
(Soomro et al.)	<ul style="list-style-type: none"> - NS-3 simulator - Packet delivery rate, energy consumption, and end-to-end latency. 	<p>TCP: data transmission is more reliable, and PDR stable with increasing network load.</p> <p>Lower end-to-end latency</p> <p>UDP: The PDR decreases with increasing network load.</p>	<p>Performance is evaluated in multiple dimensions, providing a more comprehensive view of performance.</p>	<p>The results of this article may be influenced by specific simulation parameters and scenario Settings, and therefore may not be directly applicable to all types of MANET environments.</p>
(Segara & Ramadhani, 2023)	<ul style="list-style-type: none"> - NS simulator - Data packet transmission rate and end-to-end delay 	<p>As the number of nodes increases, the PDR of TCP gradually stabilizes at a fixed value from the beginning.</p> <p>TCP has a higher packet-sending rate than UDP.</p>	<p>The performance of various routing protocols is evaluated, which provides a reference for selecting the most appropriate protocol for specific use scenarios.</p>	<p>The performance of a variety of routing protocols and algorithms is not directly compared, and direct quantitative or qualitative comparison is lacking.</p>
(Kumar & Verma, 2022)	<ul style="list-style-type: none"> - NS-3 simulator - Packet delivery ratio, end-to-end delay, routing overhead, and throughput 	<p>AOLSR/OLSR:</p> <p>Packet delivery ratio: TCP is about 20% higher than UDP.</p> <p>E2E delay: TCP is between 0.001 and 0.003 milliseconds slower than UDP.</p> <p>Routing overhead: TCP has between 10,000 and 90,000 fewer packets than UDP.</p> <p>Throughput: TCP has 2000 fewer packets than UDP.</p>	<p>The performance of TCP and UDP is compared by simulating both environments and selecting a relatively large number of metrics. It can be more clear and intuitive to understand the performance of the two under each index.</p>	<p>Performing simulation experiments in only one propagation model, it is not certain that the experimental results of AOLSR are better than those of OLSR, and the TCP is better in other models.</p>

SIMULATION SETUP AND RESULTS

Based on the summary of the existing simulation, the NS-3 simulator is selected to analyze the performance of TCP and UDP under the same variable setting, the same variable setting is the

transmission rate of 100Mbps and the simulation time of 200s. Throughput, end-to-end delay, lost packets, packet loss ratio, packet delivery rate, and bandwidth utilization are selected as the detection performance indicators in the simulation. Simulation Parameters Used are shown in Table 2.

TABLE 2. Simulation Parameters Used

Parameter	Value
Operating system	Ubuntu 20.04.6 LTS (Focal Fossa)
NS-3 version	v3.33
Simulation bandwidth	100Mbps
Simulation duration	200s
Simulation protocol	TCP, UDP
File Size	1MB, 2MB, 5MB, 10MB, 50MB, 100MB, 500MB, 1000MB
Number of Nodes	2

The rest of the simulation setup and results are described in the following sections.

1. Simulation Setup

Through the survey, NS-3 was found to be the most efficient and scalable simulator in terms of memory usage, CPU utilization, computation time, and scalability compared to other simulators (Khan et al., 2012). It is suitable for large-scale simulations because of its low memory usage, shortest computation time, and high scalability. Compared with NS-3, OPNET is more expensive and less scalable. The scalability and efficiency of OMNET++ are not as good as NS-3, and it also has certain requirements for programming ability. As a commercial software, QualNet is more expensive, less scalable, and more flexible than open-source software. NS-2 shows limitations in terms of scalability and higher resource consumption.

The NS-3 simulator was used for the simulation. Firstly, a simple end-to-end network topology was used to simulate the real network environment, and different transmission data sizes were set. Then, the simulation results are analyzed in the aspect of throughput, end-to-end delay, lost packets, packet loss ratio, packet delivery rate, and bandwidth utilization, and the performance of TCP and UDP protocols under different parameter settings is analyzed. A schematic of the end-to-end topology in Figure 5.

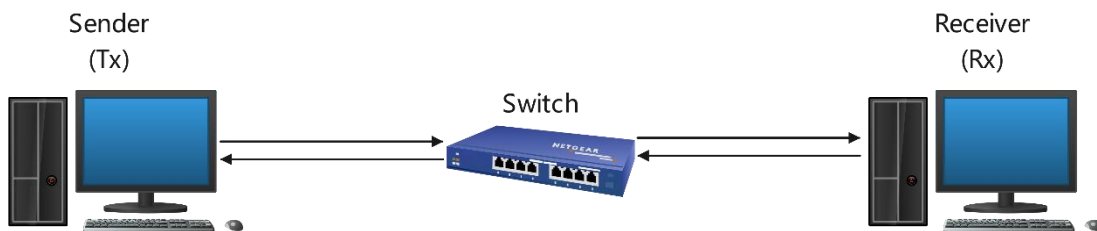


FIGURE 5. End-to-end topology

2. Simulation Results

Through the simulation of the NS-3 simulator, we obtained the following results:

i. Throughput:

Throughput represents the number of data successfully transmitted in unit time and can reflect the transmission capacity of the network. The formula for the throughput (Siahaan et al., 2022):

$$\text{Throughput} = \frac{\text{Receive (Rx) Bytes}}{\text{Time Duration}} \quad (1)$$

When the network load is low, the throughput of the UDP protocol quickly reaches a high level because the UDP protocol does not perform congestion control and directly sends data at the highest rate. However, as the file size rises, the network load further rises, and the throughput of the UDP protocol reduces significantly. Because the UDP protocol has no congestion control mechanism, the packet loss ratio rises noteworthy after the network congestion occurs, which affects the actual amount of data transmission.

Compared with UDP, when the network load is low, the throughput of TCP increases gradually over time because TCP adopts the slow-start mechanism to increase the sending rate slowly in the initial stage to avoid network congestion. However, with the increase in network load, the throughput of TCP continues to rise steadily and eventually becomes stable. This shows that congestion control mechanisms of TCP, such as congestion avoidance and fast recovery, can effectively regulate transmission rate, make the best of network resources, and avoid serious congestion. Numerical curves of throughput for TCP and UDP in Figure 6.

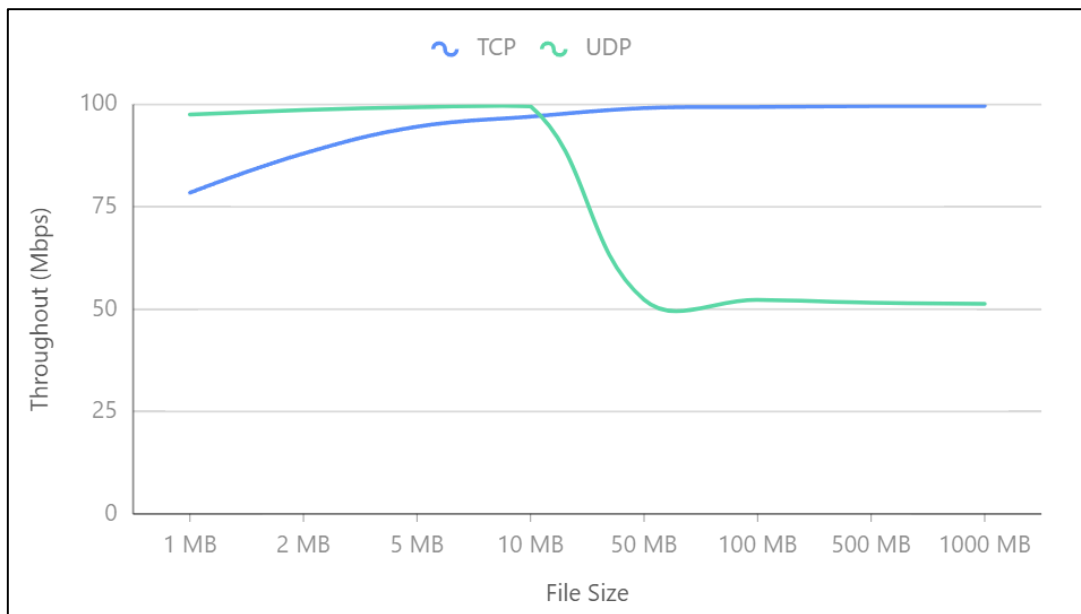


FIGURE 6. Throughput comparison between TCP and UDP

ii. End-to-End Delay:

End-to-end delay refers to the time experienced from the time a packet leaves the sender until it arrives at the receiver and can reflect the degree of real-time response. It includes sending delay, propagation delay, processing delay, and queuing delay. The formula for the End-to-End Delay (Siahaan et al., 2022):

$$\text{End - to - End Delay} = \frac{\text{Total Delay}}{\text{Number of Rx Packets}} \quad (2)$$

In the initial stage of UDP protocol, because there is no congestion control mechanism, the data packets can be transmitted quickly, so the end-to-end delay of UDP protocol is also low and stable. As the file size increases and the network load rises, the end-to-end delay of the UDP

protocol also increases significantly and rises sharply after a certain point, reaching a high value and remaining stable. This shows that the UDP protocol lacks a congestion control mechanism, and data packets may pile up in the network under high load, resulting in a significant increase in delay and a significant decrease in the transmission performance of the UDP protocol.

Compared with the UDP protocol, the end-to-end delay of the TCP protocol is relatively low and stable in the initial stage. With the rise of network responsibility, the delay of TCP protocol gradually increases to a stable level, because the TCP congestion control mechanism will reduce the sending rate and introduce retransmission when it detects network congestion, which leads to the increase of delay. However, it sacrifices part of the delay to ensure the reliable transmission of data so that the TCP protocol can maintain transmission under high load. Numerical curves of end-to-end delay for TCP and UDP in Figure 7.

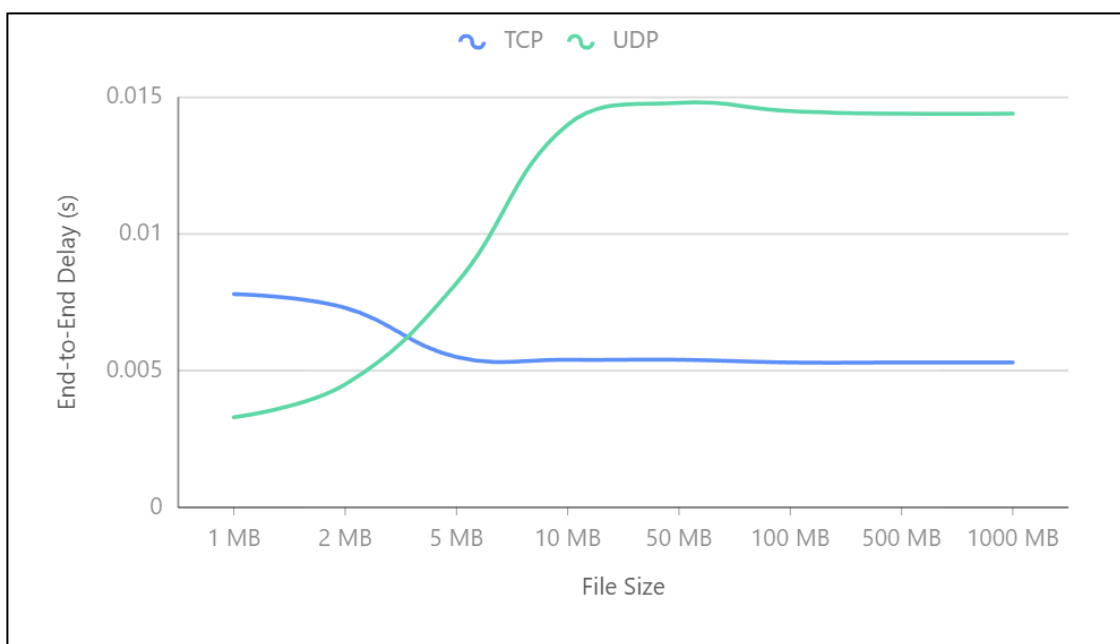


FIGURE 7. End-to-end delay comparison between TCP and UDP

iii. Lost Packets:

Lost Packet is the quantity of packets lost by the network during transmission, which can reflect the reliability of the network. The formula for the Lost Packets (Siahaan et al., 2022):

$$\text{Lost Packets} = \text{Total Packets Sent} - \text{Total Packets Received} \quad (3)$$

In the initial stage of the UDP protocol, the number of UDP protocol lost packets is also close to zero, indicating that when the network is not congested, the UDP protocol can successfully transmit data packets. As the file size rises, the network load also rises, and the number of lost packets of the UDP protocol starts to increase significantly to a sharp rise in the number of lost packets. Because UDP protocol has no congestion, it cannot adjust the transmission speed to adapt to the network conditions like TCP protocol, resulting in a rapid increase in the number of lost packets when the network load increases, showing its poor performance in congested networks.

Compared with UDP protocol, in the initial phase, the number of lost packets of TCP protocol is close to 0, which indicates that TCP protocol can smoothly transmit data packets when the network is not crowded. As the network load rises, the number of TCP lost packets starts to gradually increase, because the network starts to become congested, resulting in partial lost packets. As the network load continues to increase, TCP utilizes congestion control mechanisms to effectively decrease the number of lost packets, and the TCP keeps the number of lost packets at a low level to maintain a relatively stable transmission even under high load. Numerical curves of lost packets for TCP and UDP in Figure 8.

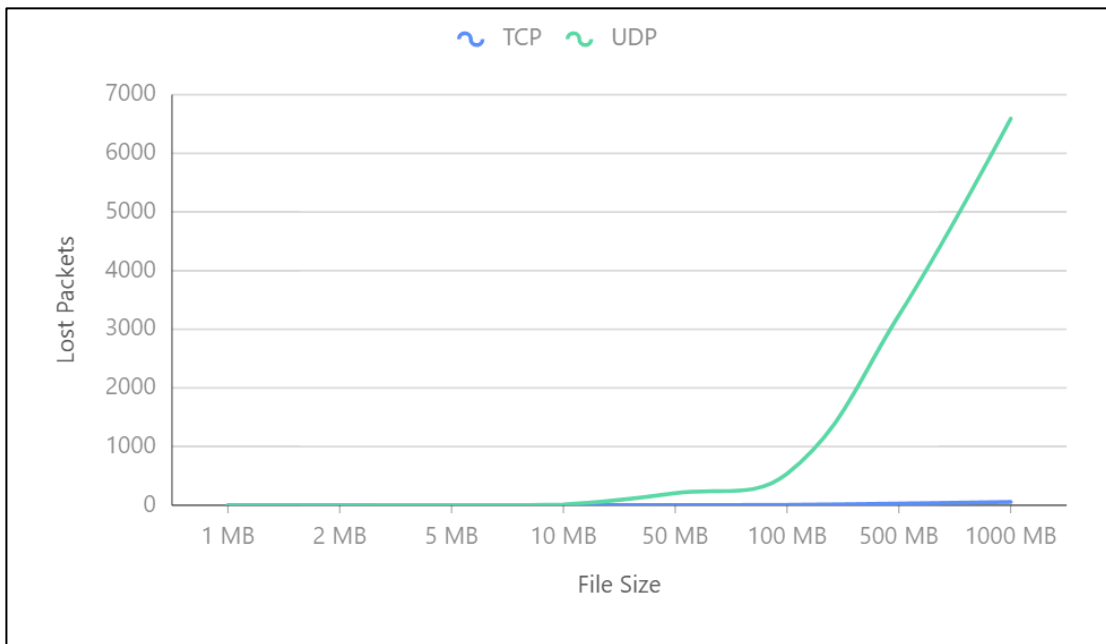


FIGURE 8. Lost Packets comparison between TCP and UDP

iv. Packet Loss Ratio:

The ratio of the number of packets lost to the total number of packets sent during data transmission is usually expressed as a percentage (Madhuri & Reddy, 2016). The formula for the Packet Loss Ratio:

$$\text{Packet Loss Ratio} = \frac{\text{Lost Packets}}{\text{Transmit (Tx) Total Packets}} \times 100\% \quad (4)$$

In the initial phase of the UDP protocol, the network load is low, and the packet loss ratio of the UDP protocol is close to 0. As the file size rises, the network load also rises, because the UDP protocol has no congestion control, so the packet loss ratio increases rapidly when the network load rises. The loss rate of UDP rises significantly from the beginning to a sharp increase in the loss rate, showing its poor performance in congested networks.

Compared with the UDP protocol, TCP protocol in the initial phase, the network load is also lower, and the packet loss ratio of the TCP protocol is close to 0. As the network load increases, the packet loss ratio of the TCP protocol starts to rise, because the network starts to become congested, resulting in partial packet loss. When the network load continues to increase, the packet loss ratio of the TCP protocol remains at a low level. This shows that TCP's congestion control mechanism effectively reduces the packet loss ratio and can maintain relatively stable

data transmission even under high load. Numerical curves of packet loss ratio for TCP and UDP in Figure 9.

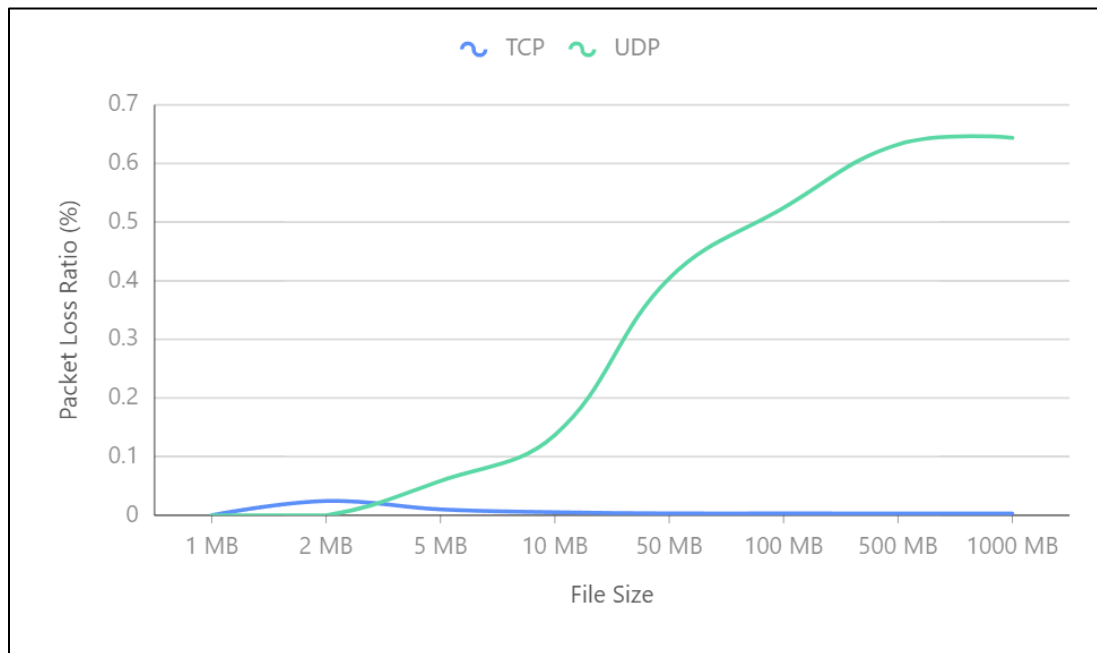


FIGURE 9. Packet Loss Ratio comparison between TCP and UDP

v. Bandwidth Utilization:

The ratio of bandwidth actually used over a period of time to the total available bandwidth is typically expressed as a percentage. The formula for the Bandwidth Utilization:

$$\text{Bandwidth Utilization} = \frac{\text{Throughput}}{\text{Available Bandwidth}} \times 100\% \quad (5)$$

The UDP protocol can initially use the available bandwidth very efficiently, achieving high bandwidth utilization from the beginning, because of the connectionless nature of the UDP protocol, which allows it to continuously transmit data without waiting for an acknowledgment. However, as the load further increases, the bandwidth utilization of UDP starts to decrease significantly. This decrease indicates that the UDP protocol cannot handle congestion well, and it will continue to send packets even when the network is congested, resulting in packet loss and reducing overall efficiency.

Compared with UDP protocol, the bandwidth utilization of TCP protocol gradually increases with the increase of load, because TCP protocol adjusts its transmission rate according to network conditions, thus improving its utilization when more bandwidth is available. As the network load continues to increase, the bandwidth utilization of the TCP protocol stabilizes at a relatively high level. This indicates that the TCP protocol can effectively utilize available bandwidth through its congestion control mechanisms, such as slow-start and congestion avoidance. Numerical curves of broadband utilization for TCP and UDP in Figure 10.

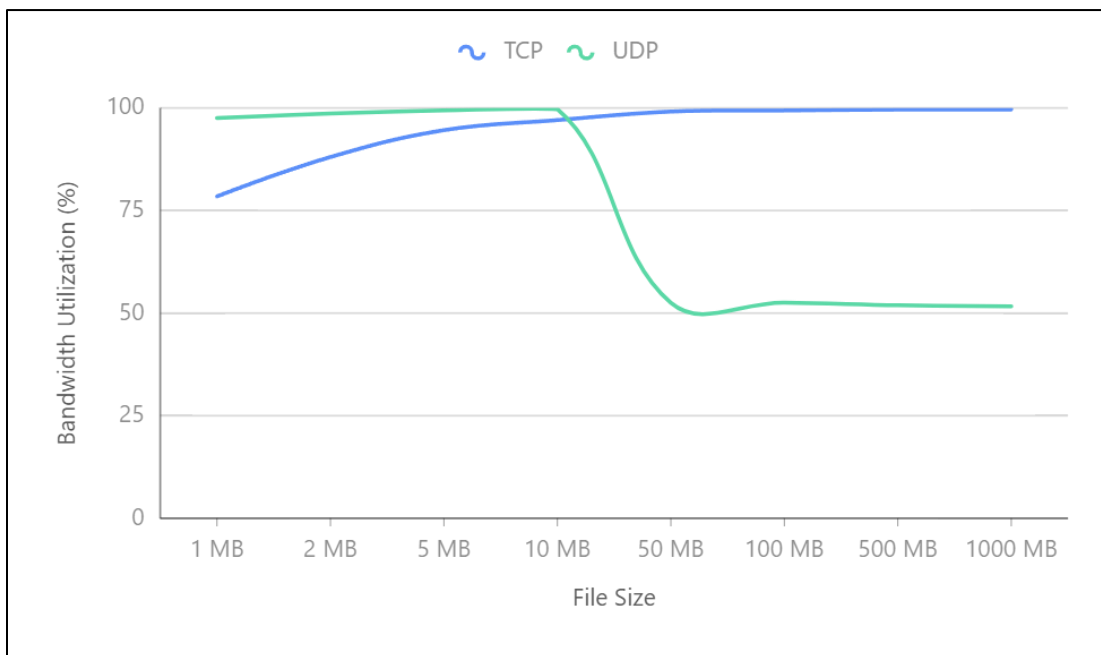


FIGURE 10. Bandwidth Utilization comparison between TCP and UDP

vi. Jitter:

Network jitter refers to the inconsistency or variation of packet arrival times during transmission. A lower jitter indicates a relatively stable packet arrival time, while a higher jitter means a less consistent packet arrival time. Formula for the Jitter (Dahmouni et al., 2012):

$$\text{Jitter} = \frac{\sum_{i=1}^n |t_i - t_{i-1}|}{n - 1} \quad (6)$$

The UDP protocol starts with a slightly higher jitter. Since the UDP protocol does not have a built-in mechanism to guarantee order and reliability, the packet arrival time may vary greatly, resulting in large jitter. As the load increases, the jitter of the UDP protocol also increases significantly, because the lack of congestion control of the UDP protocol causes the packet arrival time to vary greatly, so large jitter can adversely affect the network performance.

Compared to UDP, TCP initially has less and relatively constant jitter because TCP mechanisms such as acknowledgments and retransmissions ensure reliable and orderly delivery of packets, thus reducing the variation of packet interarrival times. As the load increases, the jitter of the TCP protocol remains relatively stable and low. Because the TCP protocol's flow control and congestion control mechanisms help maintain consistent packet delivery intervals, jitter is minimized even under heavy load. Numerical curves of jitter for TCP and UDP in Figure 11.

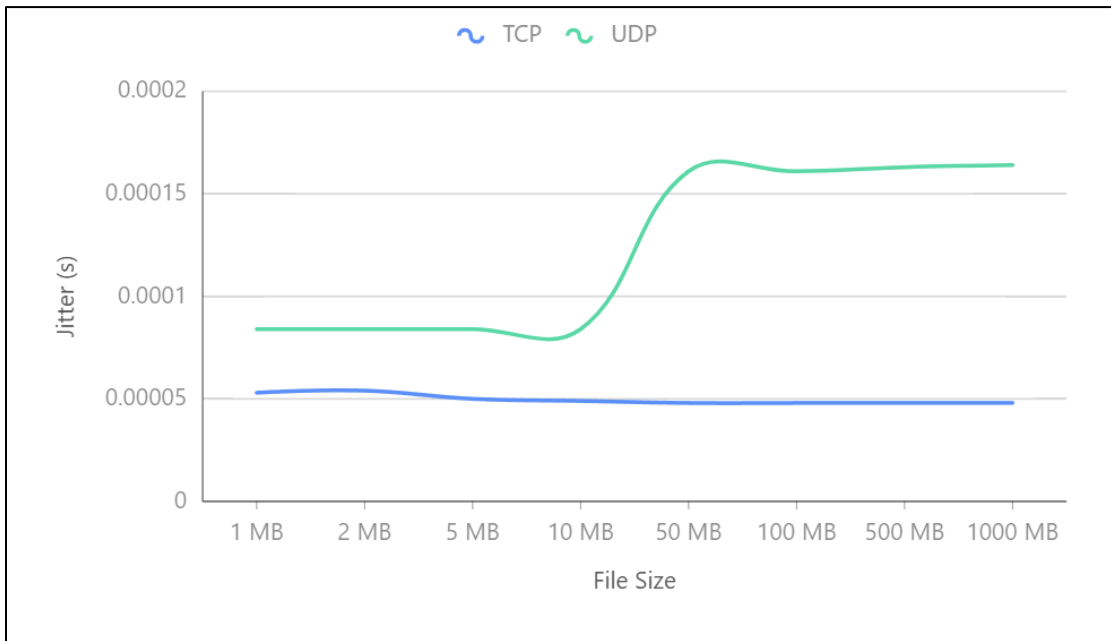


FIGURE 11. Jitter comparison between TCP and UDP

vii. Packet Delivery Ratio (PDR):

The packet delivery ratio is the ratio of the number of packets successfully received by the destination to the number of packets sent by the source, and this metric is crucial to understanding the reliability of data transmission in the network. The formula for the PDR (Venkateswara & Malladi, 2021):

$$\text{PDR} = \frac{\text{Rx Packets}}{\text{Tx Packets}} \quad (7)$$

Under the specific simulation environment conditions, the PDR of the UDP protocol remains high and stable under different conditions, indicating that UDP can successfully deliver a high proportion of data packets. Since the UDP protocol has no built-in mechanisms for ensuring reliable delivery (such as retransmission or acknowledgments), in specific environments (such as low packet loss and low congestion), UDP can still achieve high PDR due to its minimal overhead and faster transmission speed.

Compared with UDP, the PDR of TCP remains very high and stable under different conditions. This indicates that almost all packets sent by the source have been successfully delivered to the destination. TCP ensures reliable data transmission through mechanisms such as acknowledgment, retransmission, and flow control. Thus, even if packets are lost or corrupted, the TCP protocol retransmits them until they are successfully received, thus maintaining a high PDR. Numerical curves of packet delivery ratio for TCP and UDP in Figure 11.

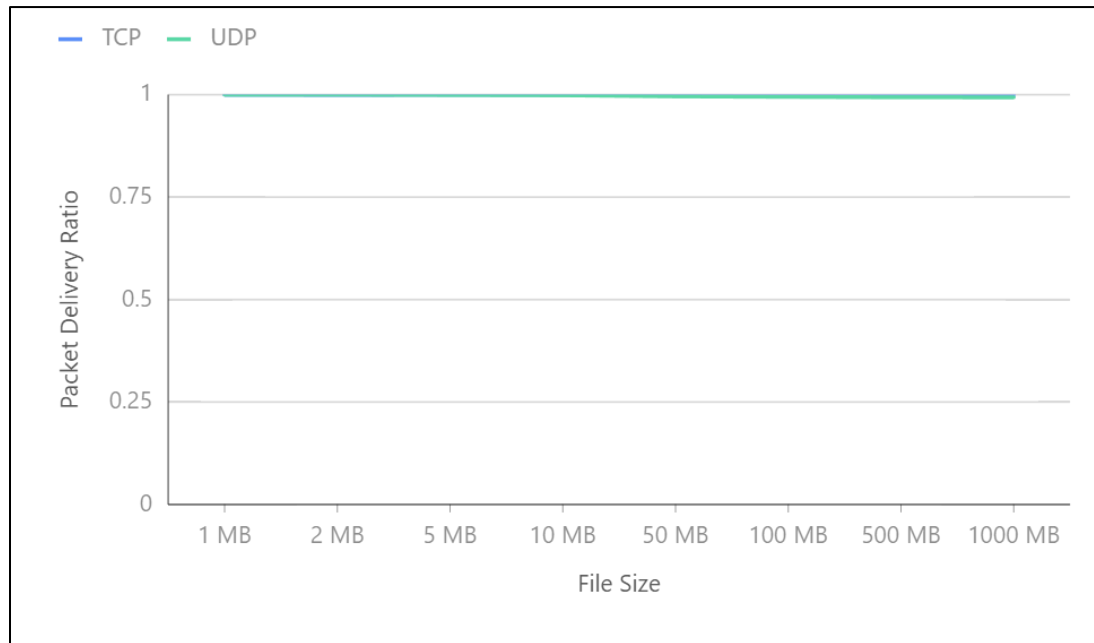


FIGURE 12. Packet Delivery Ratio comparison between TCP and UDP

The summary of the numerical results data is shown in Table 3.

TABLE 3. Simulation Result Data

File Size	Type	Throughput (Mbps)	End-to-End Delay (s)	Lost Packets	Packet Loss Ratio	Bandwidth Utilization (%)	Jitter (s)	Packet Delivery Ratio
1 MB	TCP	78.452	0.0078	0	0.0000%	78.452	0.000053	1
	UDP	97.551	0.0033	0	0.0000%	97.551	0.000084	1
2 MB	TCP	88.022	0.0073	1	0.0243%	88.042	0.000054	0.999757
	UDP	98.668	0.0045	0	0.0000%	98.668	0.000084	1
5 MB	TCP	94.583	0.0055	1	0.0100%	94.592	0.00005	0.9999
	UDP	99.35	0.0082	3	0.0586%	99.408	0.000084	0.999414
10 MB	TCP	97.043	0.0054	1	0.0051%	97.047	0.000049	0.999949
	UDP	99.579	0.014	14	0.1367%	99.716	0.000084	0.998633
50 MB	TCP	99.125	0.0054	3	0.0031%	99.127	0.000048	0.999969
	UDP	52.323	0.0148	207	0.4043%	52.536	0.000161	0.995957
100 MB	TCP	99.392	0.0053	6	0.0031%	99.395	0.000048	0.999969
	UDP	52.27	0.0145	537	0.5244%	52.546	0.000161	0.994756
500 MB	TCP	99.607	0.0053	28	0.0029%	99.61	0.000048	0.999971
	UDP	51.608	0.0144	3238	0.6324%	51.937	0.000163	0.993676
1000 MB	TCP	99.634	0.0053	55	0.0028%	99.637	0.000048	0.999972
	UDP	51.301	0.0144	6591	0.6437%	51.633	0.000164	0.993563

RESULT DISCUSSION

In this simulation, throughput, end-to-end delay, lost packets, packet loss ratio, bandwidth utilization, jitter, and packet delivery ratio of TCP and UDP protocol in the process of transferring different sizes of files are experimentally only through simple topology, and theoretical data are obtained. TCP is suitable for delay-sensitive applications that require high

data reliability, such as file transfer and web browsing. Due to its congestion control and retransmission mechanism, TCP can maintain low delay and packet loss ratio when network conditions are poor and can make full use of bandwidth to improve throughput when network conditions are good. UDP protocol is suitable for high real-time requirements and can tolerate a certain packet loss ratio of applications, such as real-time video streaming transmission, network games, etc. Due to the connectionless nature of UDP protocol, it can achieve higher throughput when the network load is low but may incur a higher delay and packet loss ratio when the network load increases.

This simulation found that the throughput of TCP protocol increases significantly with the rise of file size and becomes stable, which indicates that TCP protocol can make better use of bandwidth after network conditions improve or stabilize. On the other hand, the throughput of the UDP protocol gradually decreases and becomes stable, which is due to the increase in packet loss ratio caused by network congestion, which affects the effective throughput. The end-to-end delay of UDP protocol rises obviously with the rise in file size, which indicates that without a congestion control mechanism, UDP protocol will produce a higher delay when the network load increases. TCP's low and stable delay shows that TCP's congestion control mechanism can effectively manage the network load and avoid high delay situations. Through the simulation, it is found that the packet loss ratio of the UDP protocol rises significantly with the increase in file size, indicating that UDP will cause a large number of packet losses if there is no reliable transmission mechanism when the network is congested. The TCP packet loss ratio remains low and stable, indicating that the TCP retransmission mechanism can effectively reduce the packet loss ratio. The UDP protocol has a low packet loss ratio at the beginning, but as the file size rises, the network load also rises, making the network overwhelmed and resulting in a high packet loss ratio under heavy load conditions. The congestion control mechanism of TCP protocol helps to avoid network congestion and thus minimize packet loss, which is relatively low and stable. Although the connectionless nature of the UDP protocol has advantages in reducing latency, jitter increases when the load increases. In contrast, TCP protocol guarantees orderly and reliable transmission through its powerful packet delivery mechanism, provides more stable jitter performance, and is therefore more suitable for applications that require stable packet interarrival times. In addition, both TCP and UDP protocols can achieve a high packet delivery ratio in the specific environment of the experiment, but the context and specific application requirements must be considered. The transmission mechanism of TCP makes it highly reliable in various network conditions, while the simplicity and low overhead of UDP make it advantageous in environments with stable networks and minimal packet loss ratio. In addition, the experiment shows the difference in bandwidth utilization between the TCP protocol and the UDP protocol. The TCP protocol can use the available bandwidth more stably and effectively through a congestion control mechanism, which is suitable for application that needs reliable data transmission, while the UDP protocol is more suitable for applications whose speed priority is reliable and uses bandwidth efficiently. Although it initially achieves high bandwidth utilization, its performance degrades under high load and it does not manage congestion effectively.

CONCLUSION

This paper compares the performance of TCP and UDP by simulation in NS-3. It can be seen that the performance of TCP is better than that of UDP in aspects of throughput, end-to-end delay, and packet loss ratio. However, the performance of TCP is also affected by many external factors, among which the quality of network bandwidth and the degree of network delay are the most important factors. The quality of network bandwidth will directly affect the speed of

data transmission, and the degree of network delay will affect the real-time performance of data transmission. When these two factors are combined in different degrees, the performance of the TCP protocol will be affected in different degrees. Secondly, different network topologies also have an impact on the performance of TCP protocol. In more complex network topologies, the routing and congestion control problems faced by the TCP protocol will increase greatly. Therefore, network topology must be considered when studying TCP performance.

In short, the performance of the TCP protocol is more stable, more reliable, and more efficient than the UDP protocol, so the choice of appropriate transport protocol should be determined according to the requirements of specific applications. TCP performs well in application scenarios with reliable transmission and low latency, while UDP is more suitable for application scenarios with high real-time requirements and tolerating partial data loss.

FUTURE WORK

Future experiment needs to strengthen the optimization work, including TCP protocol (congestion control algorithm improvement, fast retransmission mechanism optimization, and parameter adjustment and adaptation) and UDP protocol (application layer retransmission mechanism, flow control and congestion avoidance, protocol expansion, and enhancement). Optimize the network configuration according to the actual needs, and improve its performance in different network environments.

In addition, future experiments could try to incorporate some targeted strategies to enhance the performance of the TCP protocol. For example, the network delay is reduced by optimizing the routing algorithm, and the data transmission efficiency is improved by improving the congestion control algorithm. The addition of these strategies can greatly optimize the performance of the TCP protocol. With the development of network technology, TCP still faces many challenges. How to maintain stable performance in various network environments is an important problem faced by TCP protocol. Therefore, future research should focus on stabilizing the influence values of each network factor to enhance the performance of TCP protocol in different network environments. This requires exploring new methods and techniques constantly in our research to find more effective solutions.

REFERENCES

- Abadleh, A., Tareef, A., Btoush, A., Mahadeen, A., Al-Mjali, M. M., Alja'afreh, S. S., & Alkasasbeh, A. A. 2022. Comparative analysis of tcp congestion control methods. 2022 13th International Conference on Information and Communication Systems (ICICS),
- AL-Dhief, F. T., Sabri, N., Latiff, N. A., Malik, N., Abbas, M., Albader, A., Mohammed, M. A., AL-Haddad, R. N., Salman, Y. D., & Khanapi, M. 2018. Performance comparison between TCP and UDP protocols in different simulation scenarios. *International Journal of Engineering & Technology*, 7(4.36), 172-176.
- Allman, M., Paxson, V., & Blanton, E. 2009. *TCP congestion control (2070-1721)*.
- Arogundade, O. R. (2023). Network security concepts, dangers, and defense best practical. *Computer Engineering and Intelligent Systems*, 14(2).
- Bertsekas, D., & Gallager, R. 2021. *Data networks*. Athena Scientific.
- Campanile, L., Gribaudo, M., Iacono, M., Marulli, F., & Mastroianni, M. 2020. Computer network simulation with ns-3: A systematic literature review. *Electronics*, 9(2), 272.
- Dahmouni, H., Girard, A., & Sansò, B. 2012. An analytical model for jitter in IP networks. *annals of telecommunications-Annales des télécommunications*, 67, 81-90.
- Haile, H., Grinnemo, K.-J., Ferlin, S., Hurtig, P., & Brunstrom, A. 2021. End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks. *Computer Networks*,

- 186, 107692.
- Khan, A. R., Bilal, S. M., & Othman, M. 2012. A performance comparison of open source network simulators for wireless networks. 2012 IEEE international conference on control system, computing and engineering,
- Kumar, P., & Verma, S. 2022. Implementation of modified OLSR protocol in AANETs for UDP and TCP environment. *Journal of King Saud University-Computer and Information Sciences*, 34(4), 1305-1311.
- Lee, C.-H., Baset, S. A., & Schulzrinne, H. 2017. Tcp over udp. In: January.
- Madhuri, D., & Reddy, P. C. 2016. Performance comparison of TCP, UDP and SCTP in a wired network. 2016 International Conference on Communication and Electronics Systems (ICCES),
- Murkomen, T. 2024. Performance, privacy, and security issues of TCP/IP at the application layer: A comprehensive survey. *GSC Advanced Research and Reviews*, 18(3), 234-264.
- Postel, J. 1980. *User datagram protocol* (No. rfc768).
- Postel, J. 1981. *Transmission control protocol* (No. rfc793).
- Segara, A. J. T., & Ramadhani, A. D. 2023. Performance Analysis of Mobile Ad-Hoc Networks Based on TCP and UDP Traffic on AODV Protocol for Warship Communication. *Journal of Systems Engineering and Information Technology (JOSEIT)*, 2(2), 53-58.
- Siahaan, B. P., Pardede, A. M., & Syahputra, S. 2022. Bandwidth Management and Web Filtering with Per Connection Queue (PCQ) Method Using Mikrotik. *International Journal of Health Engineering and Technology (IJHET)*, 1(2).
- Simon, M., Ntino, A., & Akarinwan, C. 2023. PERFORMANCE EVALUATION IN STANDARD TCP AND UDP USING NS3 SIMULATOR. *Galaxy International Interdisciplinary Research Journal*, 11(2), 224-230.
- Soomro, M. A., Channa, M. I., Zardari, B. A., Nizamani, S. Z., Arain, A. A., & Ahmed, M. Performance Evaluation of TCP and UDP Protocols Under EPLAODV Routing Protocol in Emergency Situation. 2022. *Quaid-E-Awam University Research Journal of Engineering, Science & Technology, Nawabshah. vol, 20, 29-36.*
- Stevens, W. 1997. *TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms* (2070-1721).
- Venkateswara, R. M., & Malladi, S. 2021. Improving packet delivery ratio in wireless sensor network with multi factor strategies. *International Journal of Advanced Computer Science and Applications*, 12(5).
- Yaun, G. R., Bauer, D., Bhutada, H. L., Carothers, C. D., Yuksel, M., & Kalyanaraman, S. 2003. Large-scale network simulation techniques: examples of TCP and OSPF models. *ACM SIGCOMM Computer Communication Review*, 33(3), 27-41.