DOI: 10.24996/ijs.2024.65.1.33

ISSN: 0067-2904

NS2 – Based Experimental Analysis of Throughput for TCP and UDP Traffic During Link Failure of The Network

Anshul Gupta, Kanishk Mamgain, Maninder Singh, Amit Chhabra

Department of Computer Science and Engineering, Chandigarh College of Engineering and Technology, Chandigarh, India

Received: 26/3/2022 Accepted: 12/1/2023 Published: 30/1/2024

Abstract

Link failure refers to the failure between two connections/nodes in a perfectly working simulation scenario at a particular instance. Transport layer routing protocols form an important basis of setting up a simulation, with Transmission Control Protocol and User Datagram Protocol being the primary of them. The research makes use of Network Simulator v2.35 to conduct different simulation experiments for link failure and provide validation results. In this paper, both protocols, TCP and UDP are compared based on the throughput of packets delivered from one node to the other constrained to the condition that for a certain interval of time the link fails and the simulation time remains the same for either of the protocols. Overall, this analysis is based on determining the performance of both protocols with a fixed packet size and bandwidth. This analysis, performed with the help of NS2 and XGraph, shows that the transport layer protocol, UDP acts better than TCP in terms of throughput. This opens the questions to other fellow researchers of how different metrics act in both the cases when a link failure occurs. In UDP, the throughput drops less as compared to the TCP at the time of the link failure regardless of if simulation was executed for different time periods i.e., 70,100,300,900 and 1000 seconds. The link failure interval is also varied from 10,15,20,40,350 and 440 seconds to generalize and validate the performance of the network during the interval.

Keywords: Link failure, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), NS2 Simulator, Packet Analysis, Throughput.

1. Introduction

A network is an assistance utilized to share various media files, and different relevant information between workstations, computers, and gadgets associated with an organization amid sharing other resources like keyboards, mouse, and printers. A network can be wired or wireless scenario consisting of different nodes. There are various challenges that occur while the nodes communicate with each other. Out of the different challenges of communication, Link failure is one of the most important categories that affects the overall performance of the network [1]. Link failure in a network can occur due to various reasons such as environmental threats, natural disaster, and failure of infrastructure network etc. After the link failure, packets cannot be transmitted over the designed scenario. To establish the alternative link for communication, the routing protocol would choose another alternative route to the destination. Upon failure of a link in a network, values of different Quality of Service (QoS) parameters i.e., Throughput, End to End Delay and Packet Loss may be recorded for analyzing the performance of the communication network [2].

*Email: anshulg954@outlook.com

The transmission control protocol (TCP) and user datagram protocol (UDP) protocols are transport layer protocols used for transporting packets over the network. The transport layer or the fourth layer of OSI-Model helps in the interconnection of the application layer and the network layer and is also responsible for the 'End-to-End Delivery' of the packets [3]. TCP provides reliable delivery of packets whereas UDP is a connectionless protocol, which does not provide reliability. Upon Link failure, networks can react in different ways for both these protocols. In this paper, an attempt of giving a scholarly comparison to TCP and UDP protocols based on throughput and their reaction to link failure is demonstrated.

Overall, this paper provides an extended view of what happens after a link failure occurs in a TCP and UDP based connection when a certain number of packets are sent through.
A brief description of the layout of paper is as follows: After the introduction in Section 1, Section 2 details upon the various comparison-based studies that have been conducted on TCP versus UDP. Section 3 describes the different routing protocols. Section 4 defines the NS-2 Tool. Section 5 explains the methodology of the work and discusses the scenarios in simulation and the implementation that the authors have worked on. The results in Section 6 are portrayed with visualizations of throughput over the total simulation period using XGraph followed with a the discussion in Section 7 based on these results. Finally, Section 8 concludes the paper.

2. Literary work

Different Research works have been carried out in previous years with reference to studying the performance of communication network during Link Failure. Different studies were done over TCP and UDP connection. Some of the related works are presented in this section.

In [4], authors studied TCP connection over Vehicular Ad hoc Network. This paper analyzed the packet loss during any link failure caused by the high speed of vehicles. Due to the reliable nature of TCP Protocol, the packets were again checked at the destination and the network performance was studied during route failure.

The authors in [5] studied Multi Link Failure Effects over Wide Area Network. Multi-Protocol Label Switching (MPLS) was compared with normal routing over Single and Double Failure Scenarios. Recovery time for Link failures was recorded and compared for both routing architectures.

Authors in [6] analyzed UDP and TCP protocols over network. Different parameters such as delay, energy consumption and packet delivery ratio were analyzed in different emergency situations. The researchers concluded that in emergency conditions, EPLAODV routing protocol gives better output when used with TCP.

The authors of [7] presented an exhibition assessment of UDP and TCP in MANET to evaluate the leadership of Destination-Sequenced Distance-Vector Routing (DSDV), Dynamic source routing (DSR), and Ad-hoc On-demand Distance Vector (AODV) protocols. They used NS2 to evaluate conventions execution, and the results revealed that TCP outperformed UDP in several flexibility models in terms of throughput. Similarly, the DSDV exhibited the least amount of delay in all UDP replicas.

The authors of [8] looked at many approaches to recover link failures using SDN (Software-Defined Networks) while pointing out the drawbacks of the backward, traditional networking techniques. SDN architecture raised research questions over link failures, which were also considered. Their article also looked at proactive and reactive SDN schemes using Mininet and the OpenDayLight controller, as well as simulations of tactical and data center network application scenarios.

In [9], they proposed a technique providing a detailed understanding of link failure. The procedure introduced the reliable Node ID-based resource reservation mechanism (PTN-RRP). The shortest path was then found using a criterion of approach weighted end-to-end delay. It calculated the shortest route from a specified starting point to the destination, potentially improving detection rates.

Researchers in [10] discussed various conventional Routing protocols that are well-suited to the investigation of overhead and link failure issues. SDN was suggested with a novel protection-based link failure management approach with the goal of identifying links with a high chance of failure before they fail conclusively.

For the shortest path calculation, an efficient and creative hello-based path recovery (HBPR) routing protocol was proposed. The HBPR system generates an alternate channel if a link in the network layer fails during transmission, minimizing delay time and energy consumption. The groundbreaking simplified honey pot optimization (SHPO) is then used to forecast which nodes in the network are potentially dangerous. The link failure in the Ad hoc network can reduce the overall performance of the network. As it can affect various QoS parameters of the communication scenario [11].

Link failure in MANET can be reduced by different techniques of Route migration. The researchers in [12] proposed enhancing the performance of TCP over mobile Ad hoc networks to maintain reliable communication and congestion control. But the paper has not mentioned any comparison of TCP and UDP relative to link failure. Different techniques were proposed in [13] to determine link breakages which can further improve various parameters like Packet Delivery Ratio and reduce end to end delay.

All the previous publications are related to connection failure or a comparison of TCP and UDP. However, limited research work has been conducted to compare TCP and UDP during Link Failure. To fill this research gap, authors have performed the experimental evaluation of these protocols during Link Failure over NS2 Simulator.

3. Routing Protocols

3.1 TCP

TCP (Transmission Control Protocol) is a connection-oriented, reliable protocol that supports the congestion control and the sequence numbers. It's also known as a network transmission protocol since it ensures that data is transferred in a consistent manner throughout a computer network. TCP uses an array to design all its integrity and the measures for transfer operations, such as timeouts and retransmission, in order to maintain efficiency. It is the most widely used protocol [14]. It was designed to establish a critical communication link between two nodes.

Reliability is a technique that ensures that data supplied from one node (the sender) is also received by the other node (the receiver), and in case the input is lost during communication, the sender is notified, and the data is reissued. Every byte of transmitting input is allocated a series number, and appropriate acknowledgement must be sent after the sender. The congestion control mechanism [15] of TCP is an important component. It was built on the idea that packet loss is generally relatively minimal, and that packet loss is consequently an indicator of network congestion. This is true for wired networks; however, non-congestion related packet loss may occur in mobile wireless networks. Overall, it can be considered as a consistent, directed procedure-to-procedure and summarized as a flow-adapted protocol, to name a few features.

3.2 UDP

UDP is a 'connectionless' protocol that operates at the transport layer [16]. The length, beginning of the port location, objective port location, and the checksum fields are all 8 bytes in a User Datagram Protocol header. Each of these fields has a bit depth of 16 bits. It is now troublesome due to the lack of confirmation in the information movement. As a result, an application software using UDP should completely deal with the challenges of start-to-finish communication that an association-based protocol would have dealt with. Retransmission for stable conveyance, stream management, the packetization and the reassembly, and the clog control, for example, are all possible difficulties.

Because there is no connection orientation and no destruct stage, it is rapid. As a result, it's ideal for small applications that don't require a solid connection. DNS administrations are the most well-known application of UDP. In high-speed communication where a certain amount of unreliability is tolerated, UDP performs better [17]. VoIP [18], and other application-layer standards all employ UDP as a use case for the transport protocol. In general, time-sensitive, and continuous applications, such as video traffic and audio, employ UDP because of the reason of the dropping bundles, which are preferable to delayed ones. Because of UDP's stateless nature, network applications [19], such as online gaming, employ it as a transport protocol.

3.3 Link Failure

Service Level Agreements (SLAs) are based on three simple indicators utilized by the Internet Service Providers (ISPs) that are packet loss, packet delay, and the "port" availability. Though this kind of SLA may be enough for typical Internet services (the email, normal web access, the transfer of files, and so on), it may not be adequate for new applications like Voice-over IP (VoIP). If packets are lost or delayed owing to problems caused by the optical fiber breaks, reboots of routers, various maintenance periods, and other factors, these new applications will suffer. The study performed in [20], reports that the frequency of link failures is greater than that of node failures, so ensuring the correct functioning of the network even after link failure becomes an important task. Some of the common reasons for link failures include:

1. The Router reboots, software issues, transitory equipment issues, and brief equipment or optical fiber repair procedures. Failure because of these issues last little more than a minute.
2. Equipment failures/upgrades and/or fiber cuts. These failures frequently persist longer than a few minutes.
3. One possible cause is that a router incorrectly considers an adjacency to be down when it is not. In case of router CPU being directly overloaded, the IS-IS keepalive kind of messages that are needed to detect loss of an adjacency may not be processed.
4. Hardware damage due to any reason, such as environmental interference.

4. NS2

NS2 (Network Simulator Version 2) is an open public simulation software platform based on network technology developed by UC Berkeley. In essence, it's a discrete event simulator. The simulation is powered by discrete events and has a virtual clock. It’s a popular tool for simulating communication networks. It also offers a large collection of the network and protocol objects that cover practically all elements of network technology. As a result, NS2 is the most popular simulation software used in academic research [21]. Many research centers, on the other hand, have validated the simulation's conclusions. Therefore, in this paper, NS2 is used to implement the wireless network simulation.

Tcl/Tk, OTcl, NS2, Tclcl, and other essential components are included in the NS2 software package. Tk is a graphical user development tool that may aid users in designing graphical interfaces in a visual environment; Tcl is very much an open-script language that is used to program NS2; OTcl is a Tcl/Tk-based object-oriented extension with its class structure [22]. The NS programming language, along with object-oriented simulator programming in C++ and the OTcl interpreter, is at the heart of this software bundle. NS2 supports customizable XGraph, Gnuplot, and selectable component Nam to intuitively examine and understand simulation results. NS2 can be used to implement different types of wired and wireless networks. Under any category of mobile ad hoc network i.e., MANET or VANET, NS2 Simulator can be even used to study mobility of vehicles [23]. A complete process of execution of a network simulation using NS2 is shown in Figure 1.

Figure 1: Execution of a network simulation using NS2.

5. Methodology

This section contains the simulation scenarios. Figure 2 depicts the flowchart for the complete experimental setup executed over NS-2 Simulator. The complete setup preparation i.e., the setup of each of the UDP and the TCP connection and the introduction of the link failure upon which the results are observed and generated, as depicted by the flow in Figure 2.

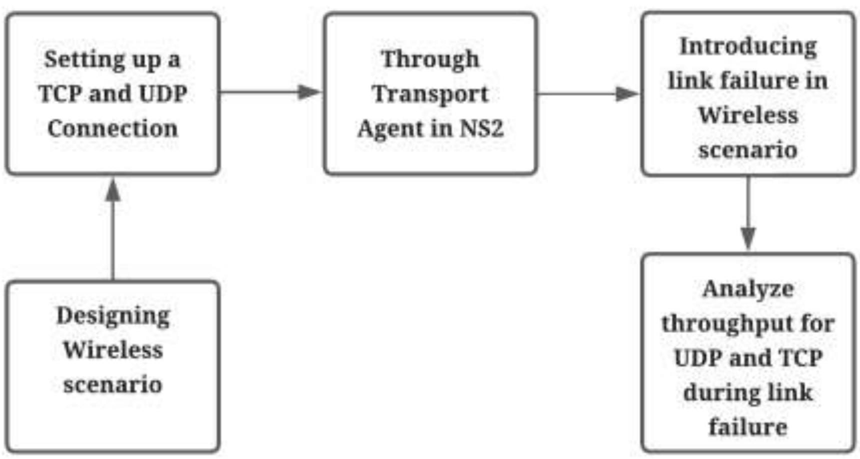

Figure 2: Flowchart of the complete setup

Table 1 represents different simulation parameters of the experiment. Initially, the nodes were created and the links were defined between them where two nodes are to be connected and the flow of data between the two nodes and the visualization of them is displayed in Figure 3 and figure 4. After this, the start and end time of the simulation was defined, that is, at what time the first packet will leave the source node and at what time the last packet will reach the destination node. As the project deals with two types of connections, TCP and UDP, there were two types of connections in the same network. The application attached with the TCP connection is FTP while with UDP, the CBR application is attached. In CBR, as the name suggests, the transfer of packet is through a constant bit rate having not much relation with FTP as FTP itself is a protocol of application layer and is used for reliable communication of data through TCP.

In setting up the TCP and UDP connection, the first step is to define the sender's transport agent and connect the sender node to this transport agent. Similarly, the transport agent is defined for the receiving node and the receiver node is attached to this receiving transport agent. Further, sending and receiving transport agents are connected and then FTP is set up over the TCP connection and CBR is set up for the UDP connection.

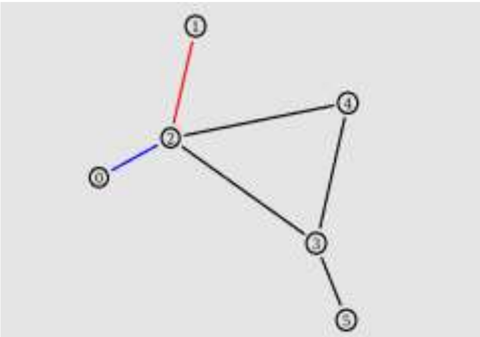

Figure 3: NAM Scenario Visualization

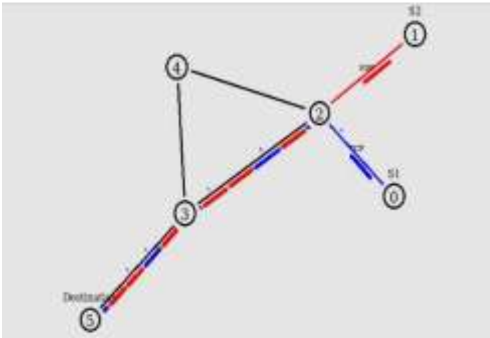

Figure 4: NAM Visualization before the link fails

Table 1: Simulation Parameters in Simulation

Parameters	Values
Simulator	NS 2.35
Number of Nodes	6 Nodes
Protocols	TCP and UDP
Speeds in Mb	1.7, 2.0, 2.7
The Packet Size(bytes)	1000
Type of Traffic for TCP	FTP
Type of Traffic for UDP	CBR
Visualizer	XGraph

After the link fails, that is when a particular link in the network through which data packets are being transferred fails. The network then automatically chooses an alternate path directed towards the destination node. Here, the link between nodes n1 and n3 disconnects. The network, instead of coming to a halt, chooses the path through node 4 and packets to reach the destination. Initially, the simulation scenario works well as depicted in Figure 4 and the packet drops start just after the link fails, shown in Figure 5.

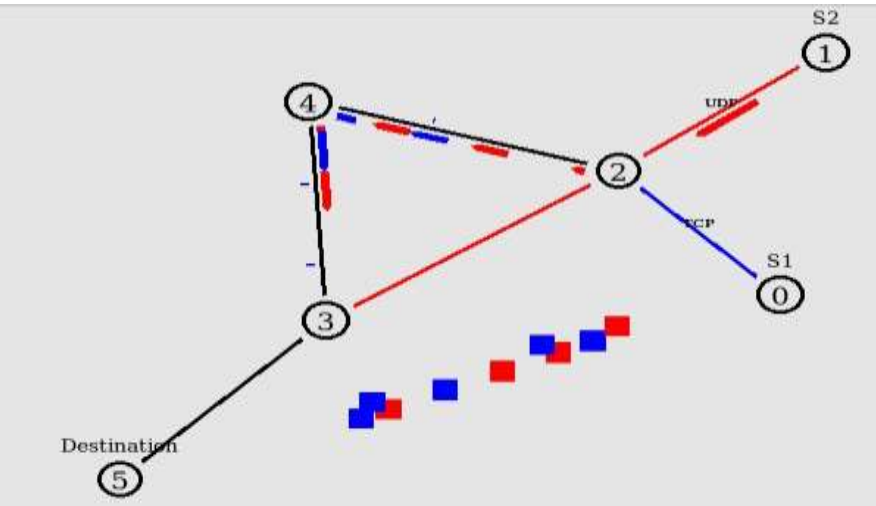

Figure 5: Link failure showing Packet Drop

For the scenario mentioned in Table 1, as the link fails, the packets that were in that link at that particular time are dropped as shown in Figure 5 and the network will choose another suitable path by itself. The new path chosen will depend on the speed and delay of the links that are available.

Performance Matrix

Throughput measures how many packets arrive at their destinations successfully. It is represented in equation (1).

$$Throughput\left(\frac{bits}{sec}\right) = \sum \frac{(N\ P)\times(A\ P\ S)}{t} \quad (1)$$

where NP denote the number of packets, APS denote the average package size, and t refers to the total time.

6. Experimental Results

Things to note while analyzing both scenarios using graph are listed below:

1- Along the x-axis, we have time in seconds.
2- Along the y-axis, we have throughput in Mbit/s.
3- UDP.tr is the trace file corresponding to UDP
4- TCP.tr is the trace file corresponding to TCP

Different simulations with the following characteristics were run and the same is depicted in Table 2. It is evident from the depictions in the Graph that there is a sharp decrease in the throughput after the link failure happens in the case of TCP. However, in the case of UDP the throughput does not drop much due to the alternate path behavior.

Table 2: Simulation Characteristics

Simulation No.	Simulation Interval	Packet Transfer Interval	Link Failure Interval	Depiction
Simulation 1	0s-100s	20-80s	30s-50s	Figure 6
Simulation 2	0s-300s	0s-300s	150s-190s	Figure 7
Simulation 3	0s-100s	0s-100s	55s-70s	Figure 8
Simulation 4	0s-1000s	0s-1000s	350s-700s	Figure 9
Simulation 5	0s-900s	0s-900s	60s-500s	Figure 10
Simulation 6	0s-70s	0s-70s	30s-40s	Figure 11

The simulation was done for different time intervals and the graph showing relation between time (in seconds) and throughput (in bits per sec) for TCP and UDP are shown in each case.

6.1 Results for Simulation 1

Figure 6 shows the graph corresponding to simulation 1 where the total simulation time was 100 seconds. It was a link between nodes 2 and node 3 as shown in Figure 4, that breaks at 30th seconds and gets reconnected at 50th seconds. This can also be seen in the graph in Figure 6, as at time = 30th seconds, a dip is seen in the throughput represented by line graph corresponding to both the protocols, which indicate a link failure. When an alternative route is established, the throughput again rises to the upper level as represented in Figure 6. It can also be observed that the drop in the throughput of UDP is far less as compared to TCP.

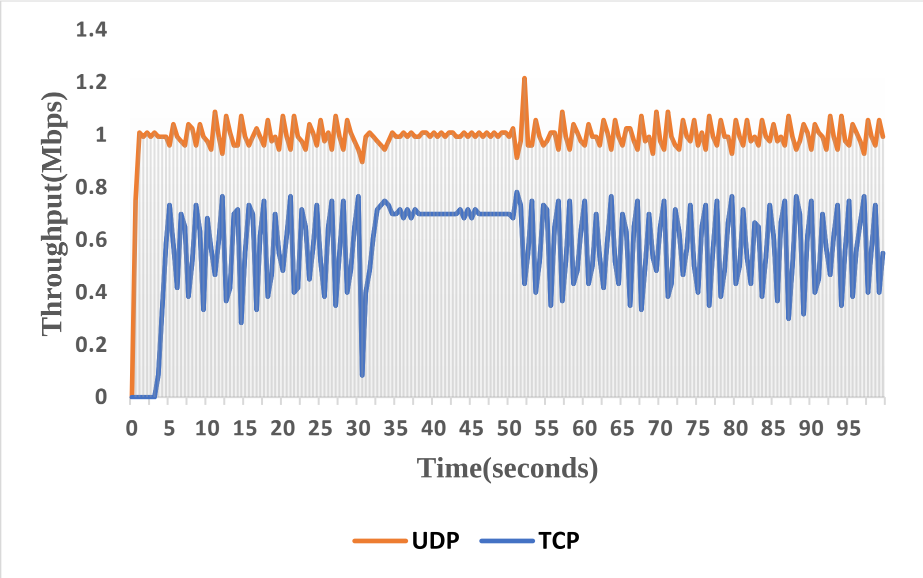

Figure 6: Simulation 1 Statistics

The simulation in Figure 6 was run for a duration of 100 seconds in which the packets were transferred for 80 seconds from 20th second to 80th second. While the link between the two nodes was down for the 30th seconds to 50th seconds, that is a total of 20 seconds.

6.2 Results for Simulation 2

Figure 7 graph corresponds to simulation 2, where the total simulation time was 300 seconds. The UDP and TCP transfer starts at 0 seconds and ends at 299 seconds. The link between nodes 2 and node 3 as shown in Figure 5 breaks at 150th second and gets reconnected at 190th seconds. This can also be seen in the graph as at time = 150 seconds, a dip indicating a drop of packets is seen in the line graph in Figure 7 corresponding to both the protocols while the throughput increases in either case after an alternate route, from other nodes than the initially transferring nodes, is found.

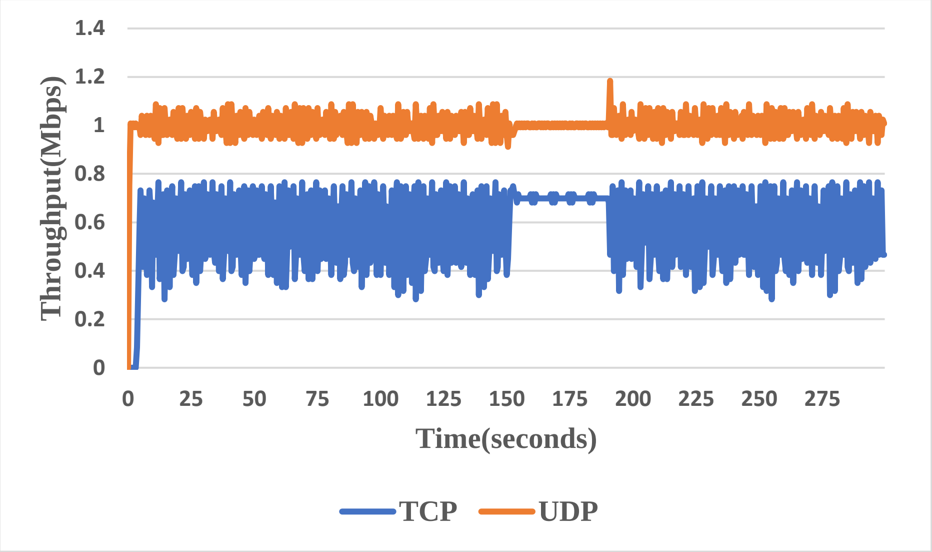

Figure 7: Simulation 2 Statistics

The simulation in Figure 7 was run for a duration of 300 seconds while the link between the two nodes was down from the 150th second to 190th second, that is for a total of 40 seconds.

6.3 Results for Other Simulation Scenarios

Similarly, the experiment was executed for different simulation times and the failure was generated at different intervals. The third simulation in Figure 8 was run for duration of 100 seconds while the link between the two nodes was down for the 55th second to 70th second, that is for a total of 15 seconds.

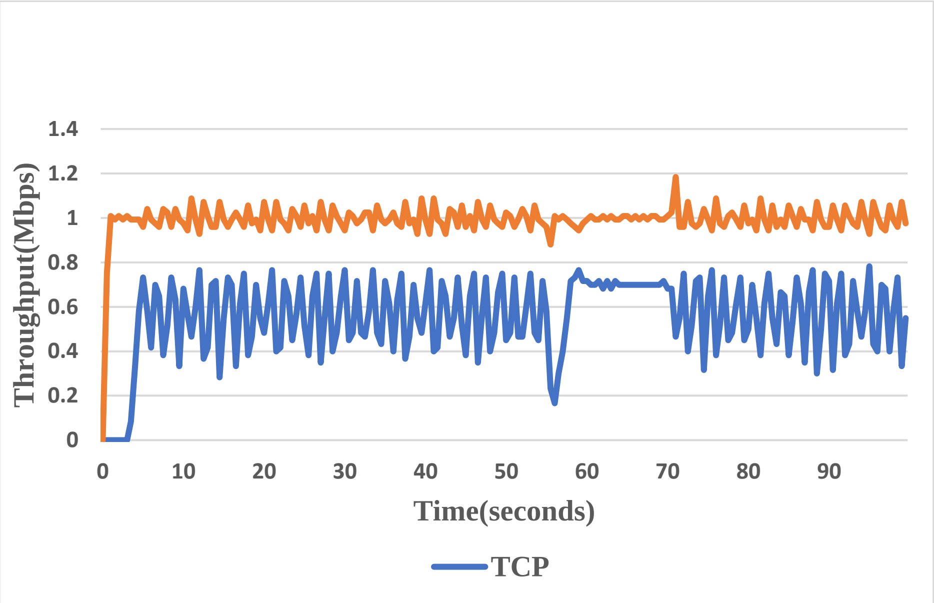

Figure 8: Simulation 3 Statistics

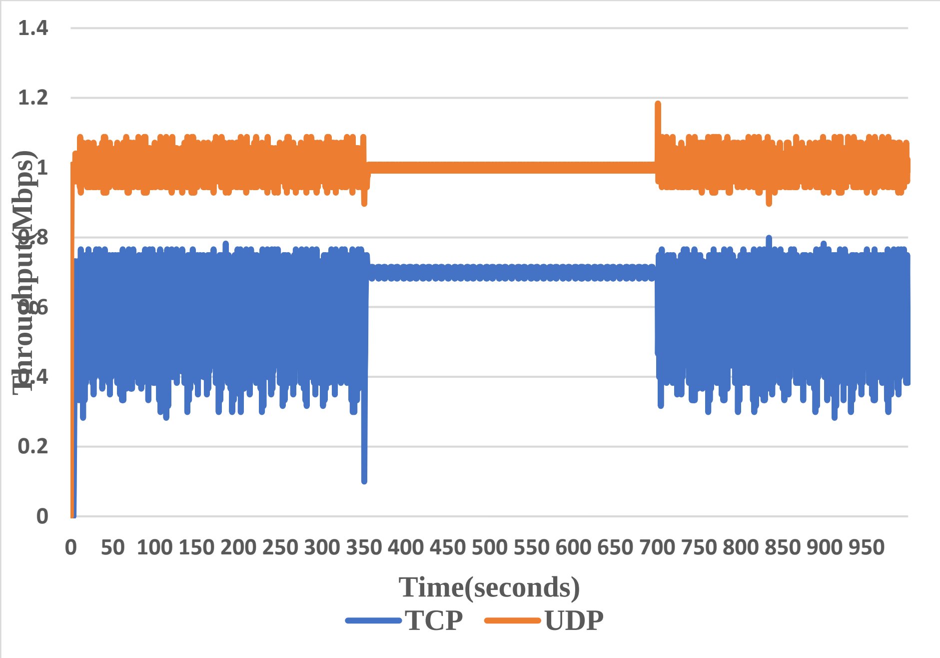

Figure 9: Simulation 4 Statistics

Simulation 4 in Figure 9 was run for duration of 1000 seconds and the link between the two nodes was down for the 350th second to 700th second, that is a total of 350 seconds.

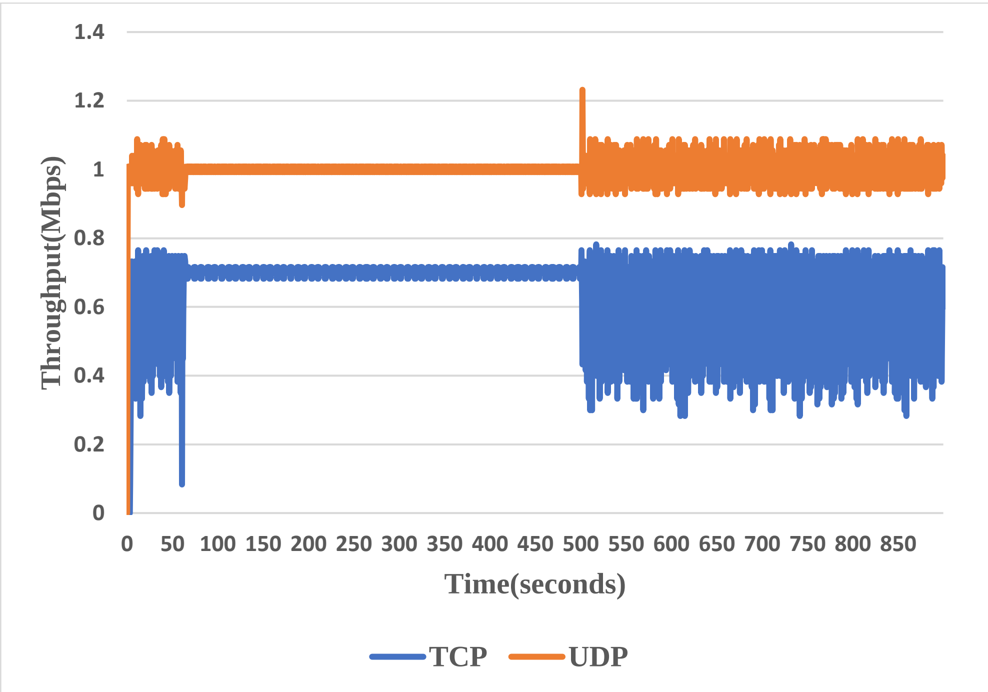

Figure 10: Simulation 5 Statistics

Simulation 5 in Figure 10 was run for a duration of 900 seconds while the link between the two nodes was down from the60th second to 500th second, that is for a total of 440 seconds.

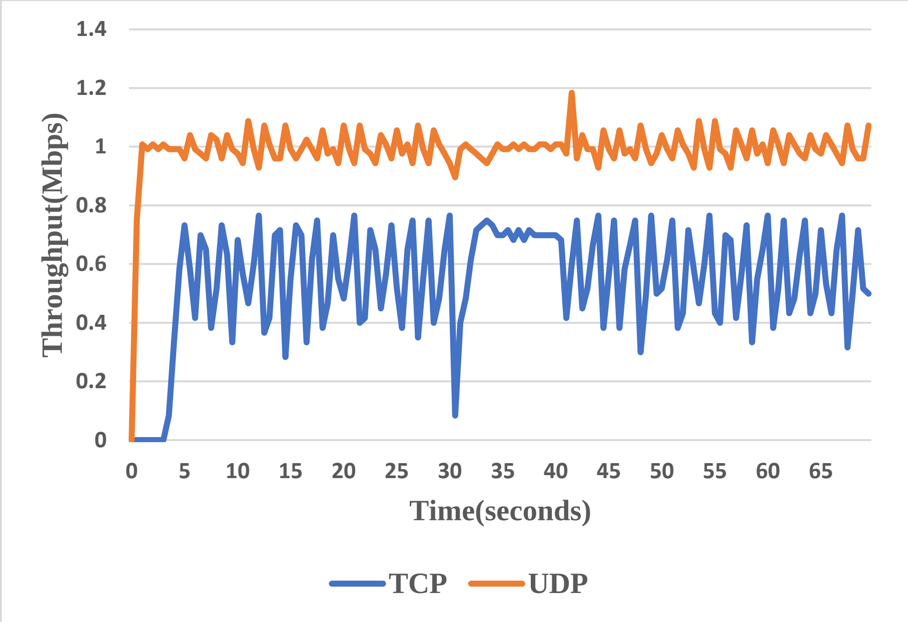

Figure 11: Simulation 6 Statistics

The sixth simulation in Figure 11 was run for duration of 70 seconds while the link between the two nodes was down for the period from 30th second to 40th second, that is for a total of 10 seconds.

7. Discussion

From the results, it can be seen that during Link Failure, throughput drops in all the six simulation scenarios whether protocol TCP or UDP is used. However, the Performance of UDP is much better than the TCP protocol. The scenario was executed for different time periods i.e., 70,100,300,900 and 1000 seconds. The link failure interval was also varied from 10,15,20,40,350 and 440 seconds to generalize and validate the performance of the network during that interval. The reason for UDP performing better in this case is because of its non-existent acknowledgement. During any congestion in the network, TCP reduces the packet transmission rate. But if UDP has packets to be sent in the network, it will occupy the major bandwidth. It results in higher throughput performance shown by UDP protocol. The results can help the researchers make decision over usage of UDP protocol over TCP protocol during link failure in a network.

8. Conclusion

From the results shown in previous section, it can be inferred that the overall throughput in the case of UDP remains more than that in TCP. Also, it can be said that in UDP, the throughput drops less as compared to the TCP at the time of the link failure regardless the scenario executed for different time periods i.e., 70,100,300,900 and 1000 seconds. After the link failure, the network automatically finds a new alternate path to the destination node till the link is reconnected without any human intervention. Thus, it can be concluded from here that UDP acts better in terms of throughput when a link failure occurs for a particular interval. In the future work, link failure experimental analysis can be done on different IEEE Standards.

9. Conflict of Interest

The authors declare that there has not been any conflict amongst the authors in the work stated.

10. Availability of data and material (data transparency)

No external use of the data has been made during the course of this research. The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

11. Funding Information

The authors did not receive support from any organization for the submitted work.

References

[1] H. Seddiqi and S. Babaie, "A new protection-based approach for link failure management of software-defined networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 4, pp. 3303–3312, 2021.

[2] B. H. Khudayer, M. Anbar, S. M. Hanshi, and T.-C. Wan, "Efficient route discovery and link failure detection mechanisms for source routing protocol in mobile ad-hoc networks," *IEEE Access*, vol. 8, pp. 24019–24032, 2020

[3] Seetha Ramanjaneyulu, K. Annapurna, and Y. Ravi Sekhar, "Performance of UDP in comparison with TCP in vehicular communication networks," in *Evolutionary Computing and Mobile Sustainable Networks*, Singapore: Springer Singapore, 2022, pp. 899–910.

[4] A. R. Ahmed, M. Ould-Khaoua, and M. Meribout, "The impact of routing protocols on the performance of TCP in VANETs," in *2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS)*, 2021.

[5] W. Gray, A. Tsokanos, and R. Kirner, "Multi-link failure effects on MPLS resilient fast-reroute network architectures," in *2021 IEEE 24th International Symposium on Real-Time Distributed Computing (ISORC)*, 2021

[6] Soomro, Muhammad Ali, M. Ibrahim Channa, Baqir Ali Zardari, Shah Zaman Nizamani, Aijaz Ahmed Arain, and Mukhtiar Ahmed,"PERFORMANCE EVALUATION OF TCP AND UDP PROTOCOLS UNDER EPLAODV ROUTING PROTOCOL IN EMERGENCY SITUATION," *Quaid-E-Awam University Research Journal of Engineering, Science & Technology, Nawabshah.vol.* 20, no.01, pp. 29-36, 2022

[7] F. T. AL-Dhief, N. Sabri, M. S. Salim, S. Fouad, and S. A. Aljunid, "MANET routing protocols evaluation: AODV, DSR and DSDV perspective," *MATEC Web Conf.*, vol. 150, p. 06024, 2018.

[8] J. Ali, G.-M. Lee, B.-H. Roh, D. K. Ryu, and G. Park, "Software-defined networking approaches for link failure recovery: A survey," *Sustainability*, vol. 12, no. 10, p. 4255, 2020.

[9] P. Srividya, L. Nirmala Devi, and A. Nageswar Rao, "A trusted effective approach for forecasting the failure of data link and intrusion in wireless sensor networks," *Theor. Comput. Sci.*, vol. 941, pp. 1–13, 2022.

[10] R. Mohammadi and R. Javidan, "EFSUTE: a novel efficient and survivable traffic engineering for software defined networks," *J. Reliab. Intell. Environ.*, vol. 8, no. 3, pp. 247–260, 2022.

[11] S. Kumar, "Prediction of node and link failures in mobile ad hoc network using hello based path recovery routing protocol," *Wirel. Pers. Commun.*, vol. 115, no. 1, pp. 725–744, 2020.

[12] P. Chaudhary and S. Kumar, "Comparative study of TCP variants for congestion control in wireless network," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, 2017

[13] S. Venkatasubramanian, A. Suhasini, and C. Vennila, "Efficient multipath zone-based routing in MANET using (TID-ZMGR) ticked-ID based zone manager," *Int. j. comput. netw. appl.*, vol. 8, no. 4, p. 435, 2021.

[14] A. Deshpande, "Efficient TCP Performance in Ad-Hoc Networks Using Novel Hybrid Adhoc Collective Space Direction - Genetic Algorithm," *INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING & TECHNOLOGY*, vol. 12, pp. 314–327, 2021.

[15] J. Himeedah, A. Hamouda, M. Taher, and S. Sati, "TCP congestion analysis of wired and wireless links using NS3," in *2022 IEEE 2nd International Maghreb Meeting of the*

Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), 2022.

[16] N. M. Garcia, F. Gil, B. Matos, C. Yahaya, N. Pombo, and R. Goleva, “Keyed User Datagram Protocol: concepts and operation of an almost reliable connectionless transport protocol,” *IEEE Access*, vol. 7, pp. 1–1, 2019.

[17] B. J. Santhosh Kumar and V. S. Sanketh Gowda, “Detection and prevention of UDP reflection amplification attack in WSN using cumulative sum algorithm,” in *2022 IEEE International Conference on Data Science and Information System (ICDSIS)*, 2022.

[18] O. P. Roy and V. Kumar, “A survey on voice over internet protocol (VoIP) reliability research,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1020, no. 1, p. 012015, 2021.

[19] Tahir, Muhammad Naeem, and Marcos Katz, "ITS performance evaluation in direct short-range communication (IEEE 802.11 p) and cellular network (5G)(TCP vs UDP)," In *Towards Connected and Autonomous Vehicle Highways*, pp. 257-279. Springer, Cham, 2021.

[20] P. Srivi, L. N. Devi, and N. Rao, “A trusted approach for prediction of data link failure and intrusion detection in wireless sensor networks,” *Research Square*, 2021.

[21] I. A. Aljabry and G. A. Al-Suhail, “A survey on network simulators for vehicular ad-hoc networks (VANETS),” *Int. J. Comput. Appl.*, vol. 174, no. 11, pp. 1–9, 2021.

[22] A. K. Walia, A. Chhabra, and D. Sharma, “Comparative analysis of contemporary network simulators,” in *Innovative Data Communication Technologies and Application*, Singapore: Springer Nature Singapore, 2022, pp. 369–383.

[23] K. T. Chui *et al.*, “Traffic accident prevention in low visibility conditions using VANETs Cloud environment,” *Int. j. cloud appl. comput.*, vol. 12, no. 1, pp. 1–21, 2022.