Comparative Analysis of Protocol Performance in different Topologies: Finding Optimal Packet Size and Sending Interval

*Project Level Sought: A

Zhenhan Lin
dept. Electrical and Computer Engineering
University of Florida
Gainesville, United States
zh.lin@ufl.edu

Kaiyi Lei
dept. Electrical and Computer Engineering
University of Florida
Gainesville, United States
kaiyi.lei@ufl.edu

***Abstract*—This report presents an comparitive analysis of network performance using the Mininet tool, a crucial instrument for network examination. The focus of the study is on evaluating the impact of two key parameters: packet size and packet transmission interval, on the performance across various network topologies and protocols. Specifically, we investigate how these parameters influence network behavior in six different topologies with the same sending node and destination node under three distinct protocols. Our findings reveal that a packet size of 64k bytes and a transmission interval of at least 0.0001 seconds optimally ensure successful data transmission. This configuration results in a 100% success rate in transmission tests across all topology types, maintaining a 0% packet loss rate when operating under each of the three evaluated protocols.**

***Index Terms*—topology, protocol, packet size, transmission interval**

I. Introduction

In the realm of network engineering and analysis, the intricate interplay between network topologies and protocols plays a pivotal role in defining the overall performance and reliability of a network. This report delves into a study using the Mininet tool [3], an established platform for network emulation and analysis, to scrutinize the performance metrics of diverse network topologies and protocols under various conditions. Our investigation focuses on six distinct network topologies: mesh, linear, tree, star, ring, and hybrid [4], each presenting unique characteristics and challenges in network communication. Alongside these topologies, the study examines three widely-used network protocols: UDP (User Datagram Protocol), TCP (Transmission Control Protocol), and SCTP (Stream Control Transmission Protocol) [2], each known for their distinct mechanisms in data transmission.

The performance of these network arrangements is quantitatively assessed based on four critical metrics: packet loss rate, transfer time cost, latency/RTT (Round-Trip Time), and throughput. These metrics are instrumental in evaluating the efficiency, reliability, and overall capability of a network structure in handling data under varying protocols and topological arrangements. By analyzing these parameters, the report aims to shed light on the optimal configurations and conditions for each topology and protocol, thereby contributing to more informed decisions in network design and optimization. The ensuing sections present a detailed analysis of our findings, offering a nuanced understanding of how different network setups fare in real-world scenarios.

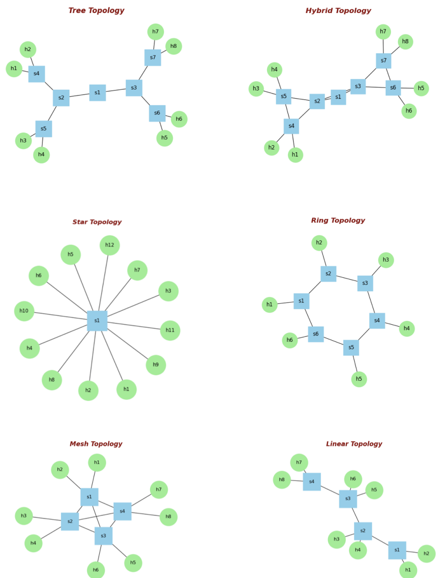

Fig. 1: Topology Visualization

II. Experiment Setup

A. Topology design

We have modeled and analyzed six key types of network topologies, each presents unique connectivity patterns and serves different networking needs. To streamline the experimental process and focus on comparing network performance under different packet sizes and sending intervals, we have fixed the sending node and the destination node as 'host 1' and 'host 3' for each topology in each of our experiments. This approach ensures consistency across different network topologies, allowing for a more direct comparison of their respective performances under similar conditions. The parameters we have set as constants and the sets of parameters we are comparing are organized as follows:

Parameter	Value
packets number	500
time out	1s
intervals(ms)	0.01, 0.05, 0.1, 0.5, 1
message size(bytes)	2, 4, 8, 16, 32, 64, 128

TABLE I: Parameter Setting

In our study, we have developed Python scripts to automatically run each experiment and compute key network performance metrics, including packet loss rate, transfer time cost, latency/RTT (Round-Trip Time), and throughput. The specific methodologies employed for these calculations are detailed below:

$$\text{Transfer time cost} = \text{end time - start time}$$

$$\text{Throughput} = \frac{\text{Transferred data size}}{\text{Transfer time cost}}$$

$$\text{Packet loss rate} = 1 - \frac{\text{recieved packets}}{\text{total send packets}}$$

$$\text{Average latency} = \frac{\text{total latency}}{\text{recieved packets}}$$

III. Results and Analysis

A. Comparison among topologies and protocols

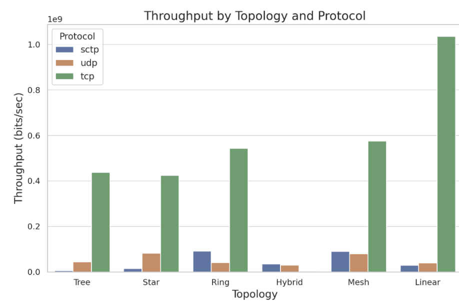

Fig. 2: Throughput by topology and protocol

The Fig.2 presents the throughput results for three protocols across various topologies, under the conditions of a 64k packet size and a 0.00001s sending interval. Notably, the TCP protocol consistently shows the highest throughput across all topologies, with its peak performance in the Tree topology. In contrast, SCTP and UDP generally exhibit lower throughputs, especially in the Star and Mesh topologies where the disparity in performance is most pronounced.

These findings indicate that, for requirements of high-speed data transmission and low latency, the TCP protocol offers superior performance in a range of network topologies. However, considering the applicability and advantages of different topologies in network design, the choice of the right protocol remains a crucial decision based on specific application scenarios and performance requirements.

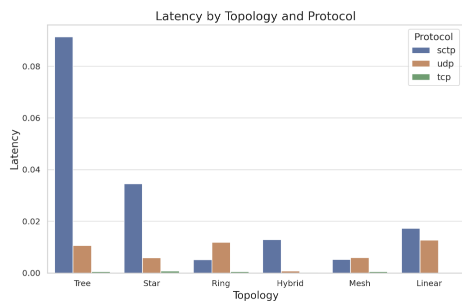

Fig. 3: Latency by Topology and Protocol

The Fig.3 presents the latency result with the same condition in the experiment of Fig.2. As can be seen in the Fig.3, the latency of the SCTP protocol is consistently the highest across all topologies, except for the Ring and Mesh topologies. In contrast, the TCP protocol consistently demonstrates the lowest latency.

Additionally, from the chart, it is evident that the latency performance varies significantly between different topologies. The Tree topology, for instance, shows a notably higher latency for SCTP compared to UDP and TCP. On the other hand, the Star topology [1] exhibits relatively lower latency values across all protocols, suggesting that this topology may be more efficient for these protocols. The variations in latency among the topologies underscore the importance of choosing the appropriate network structure depending on the protocol used and the specific performance requirements.

B. Relationship among parameters and statistics

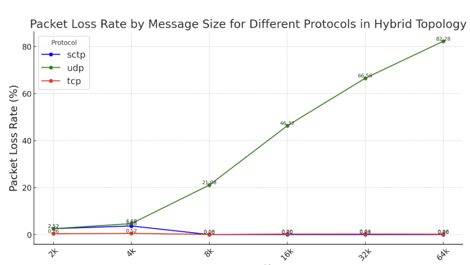

Fig. 4: Packet loss rate by message size in Hybrid Topology

In Fig.4, the experiment illustrates the packet loss rate in a Hybrid Topology under various message sizes.Notably, the UDP protocol shows a significant increase in packet loss as message size grows. In contrast, the TCP protocol maintains a remarkably low packet loss rate throughout, never exceeding 0.52% (for 4k message size). SCTP presents an intermediate performance, with its highest packet loss rate observed at 3.68% for 4k message size.

The experiment underscores the critical interplay between protocol selection and message size in Hybrid Topology networks. It indicates that in practical network implementations, the choice of both the protocol and optimal message size must be carefully considered to ensure reliable data transmission.

C. Relationship among parameters and statistics

In Fig.5, which is a result from linear topology, the optimal choice of sending interval and packet size varies notably between protocols. A key observation is the heightened sensitivity of TCP to changes in the sending interval, as evidenced by significant fluctuations in throughput with even minor variations in the interval. For TCP, the most efficient interval identified is the smallest one tested, 1e-05 seconds, particularly effective for the largest packet size of 128k. Notably, TCP is the only protocol capable of successfully handling packets of this size, indicating its suitability for scenarios requiring frequent transmission of large-scale data.

In contrast, SCTP and UDP also respond to changes in sending intervals, but their throughput variations are not as pronounced as in TCP. A moderate interval, around 0.0001 seconds, along with medium packet sizes such as 16k or 32k, appears to offer a balanced throughput and stability for these protocols.

Therefore, the selection of protocol, sending interval, and packet size in linear topology should be tailored to the network's specific requirements and characteristics. TCP is appropriate for scenarios demanding the transmission of very large packets at high frequencies, while SCTP and UDP are better suited for handling medium-sized packets, especially where extremely large packet sizes are not necessary.

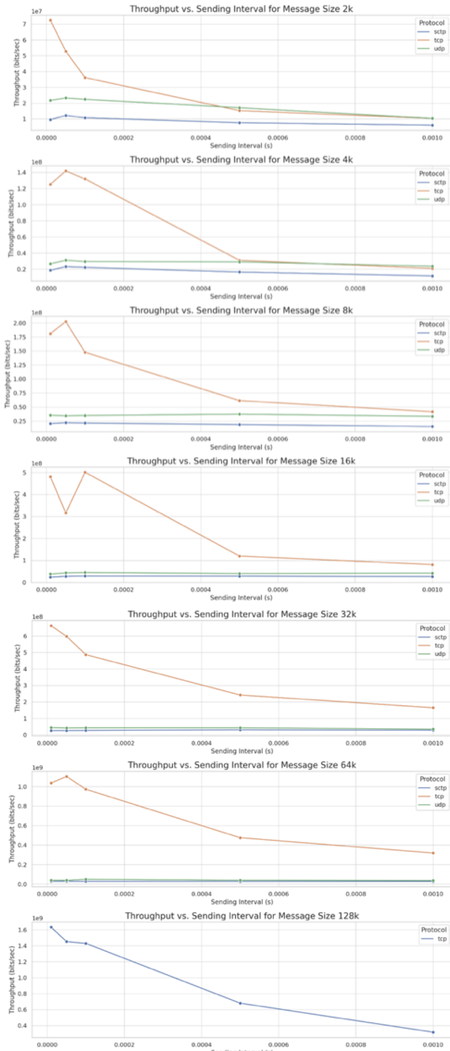

Fig. 5: Throughput by different intervals and message size

IV. Conclusion

In our experiment, which utilized relatively simple topologies, we observed that TCP is capable of transmitting larger packet sizes compared to SCTP and UDP. This capability of TCP to handle significantly larger data packets underscores its suitability in scenarios where large volume data transmission is essential.

Further, both TCP and SCTP demonstrate markedly higher transmission reliability than UDP when dealing with larger message sizes. TCP stands out with its lower latency and higher throughput, offering a compelling combination of efficiency and reliability. This distinction becomes particularly crucial in applications where the speed and consistency of data transfer are critical.

UDP, on the other hand, shows relatively better reliability for small packet sizes, particularly at 2k. However, as the packet size increases, its average packet loss rate rises significantly. Despite this, UDP consistently maintains the lowest latency across all tested sizes and is not as sensitive to changes in sending intervals as the other protocols. This characteristic makes UDP particularly suitable for the transmission of smaller packets where minimal latency is prioritized.

Based on the data from our experiment, the optimal packet size and sending interval for TCP, UDP, and SCTP are as follows. For TCP, the optimal packet size is 128k with a sending interval of 1e-05 seconds. This combination ensures maximum throughput and efficient handling of large packets. For SCTP, the ideal packet size appears to be in the medium range, such as 16k or 32k, with a sending interval around 0.0001 seconds, balancing throughput and stability. Meanwhile, UDP performs best with smaller packet sizes like 2k, where it maintains low latency and acceptable reliability, making shorter sending intervals preferable.

References

[1] Deepak S, Anandakumar H, Pavithra S, et al. Performance Analysis of Star Topology for Small Networks Using Riverbed[C]//2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, 2022, 1: 2108-2111.

[2] Ong L, Yoakum J. An introduction to the stream control transmission protocol (SCTP)[R]. 2002.

[3] Kumar D, Sood M. Software defined networks (SDN): experimentation with Mininet topologies[J]. Indian Journal of Science and Technology, 2016, 9(32): 1-7.

[4] Shozi T, Mudali P, Adigun M O, et al. A data plane multi-path load balancing mechanism for hybrid software defined networks in different topologies[C]//2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD). IEEE, 2019: 1-6.