A QoS-aware Mechanism for Reducing TCP Retransmission Timeouts using Network Tomography

Jingfu LI*
College of International Education,
Huanghuai University, Zhumadian 463000,
Henan Province, China

***Abstract*—A wide range of web-based applications uses the Transmission Control Protocol (TCP) to ensure network resources are shared efficiently and fairly. As wired and wireless networks have become more complex, various end-to-end Congestion Control (CC) schemes have been developed, offering solutions through their proposed TCP variants. Network tomography, a powerful analytical tool, offers a unique perspective by measuring end-to-end performance to estimate internal network parameters, including latency. This estimation capability proves valuable, especially in cases where precise protocol performance evaluation is essential. TCP protocol can be improved significantly by properly estimating RTT time. It has resulted in better network conditions and improved reliability, as well as a higher level of user satisfaction. In this study, we propose a method to infer the link delay using network tomography and then adjust the RTT based on the delay estimation obtained in the previous step. Simulation results performed using the NS2 software show that the proposed method significantly improves the TCP protocol's Round-Trip Time (RTT) estimation by more than 15%. It reduces congestion, improves information transfer efficiency, and ensures the highest level of service in the network.**

***Keywords*—Latency; network tomography; end to end; depending on the probe**

I. Introduction

The landscape of networking technologies has evolved significantly over the years, witnessing advancements that span multiple generations, including the emergence of 5G and the anticipation of 6G. The need for efficient data transfer mechanisms becomes increasingly pronounced as wired and wireless networks expand in complexity and scale. The Transmission Control Protocol (TCP) has long been a cornerstone of network communication, ensuring reliable and adaptive data delivery in various applications [1, 2]. While high-speed networks have reached gigabit speeds, mobile wireless access networks have led to a proliferation of mobile hosts connected to the Internet via slow wireless links [3]. Further, the challenging characteristics of wireless links, such as the high packet loss rates or delays resulting from various factors, such as link-layer retransmissions or handoffs between connection points to the Internet, have posed significant challenges to Internet transport protocols [4, 5]. Today's Internet applications require a reliable mechanism to transfer data due to increasing performance requirements. TCP is extensively used as the transport protocol in many applications due to its ability to adapt to the properties of the network and its robustness in the face of many types of failures [6, 7]. However, the closed nature of legacy switches, which do not provide accurate visibility of network events, has limited the improvement of the performance of applications that rely on TCP [8].

In this rapidly evolving landscape of complex systems and advanced technologies, research has been directed toward optimizing control strategies and enhancing performance across various domains. The integration of hierarchical optimization and fuzzy logic has yielded promising results in addressing challenges posed by time-varying delays and disturbances in discrete large-scale systems [9, 10]. Machine learning techniques have also emerged as a powerful toolset for analyzing and managing network dynamics. Notably, studies have delved into the analysis of Android ransomware using hybrid approaches [11] and explored machine learning-based network slicing for efficient 5G network management [12]. Additionally, predictive models and probabilistic neural networks have been harnessed to forecast idle slot availability in wireless local area networks [13], while innovative cell designs have been introduced to enhance data collection efficiency in green IoT networks [14]. The integration of machine learning into data conditioning and forecasting methodologies has showcased its potential in optimizing well-pad operations [15, 16]. In the rapidly evolving landscape of complex systems and advanced technologies, the fusion of association rule mining and urban public transportation assumes paramount importance. This synergy facilitates data-driven decision-making, offering insights into commuter behavior, traffic patterns, and service optimization. As urban areas continue to grow, leveraging these tools empowers city planners and transport authorities to navigate the complexities of modern urban mobility, fostering efficiency, sustainability, and improved quality of life for urban dwellers [17, 18]. Moreover, sustainable energy technology has embraced novel concepts, such as power harvesting through ambient vibrations and capacitive transducers [19]. This compilation of research endeavors exemplifies the diverse and impactful directions that contemporary studies are exploring, aiming to unlock new frontiers of knowledge and practical application.

TCP uses the fast retransmit mechanism to trigger retransmissions following the receipt of three consecutive duplicate acknowledgments (ACKs) [20]. As a backoff mechanism, the TCP retransmission timer expires if the TCP sender does not receive ACKs for a certain period of time. Upon expiration of the retransmission timer, the TCP sender retransmits the first undelivered segment, assuming it is lost in the network [21]. Because retransmission timeouts (RTOs) can indicate that the network is heavily congested, the TCP sender resets its congestion window to one segment and gradually increases it in accordance with the slow start algorithm. Nevertheless, suppose the RTO occurs spuriously, and segments are still outstanding in the network. In that case, a false slow start may damage the potentially congested network by injecting additional segments at an increased rate [22]. One of the most crucial aspects of such a mechanism is how long after sending a package and receiving a receipt, the timeout must be announced. Network tomography offers the opportunity to actively measure link-level characteristics such as delay and loss on end-to-end paths at the link level [23]. Most network tomography techniques developed to date are based on one-way measurements, requiring cooperation from sending and receiving hosts. Consequently, the paths over which these techniques can be applied are severely limited [24]. This paper proposes a method for estimating link delay using network tomography, followed by adjusting the RTT in accordance with the delay value obtained in the previous step. Finally, the results from the study are used to improve the TCP retransmission mechanism to reduce latency and improve performance. The main contributions of this paper are summarized as follows:

- We propose a novel method that leverages network tomography to estimate link delays and adjust RTT values, significantly improving TCP protocol performance.
- Simulation results conducted using NS2 software demonstrate that our approach enhances TCP RTT estimation by over 15%, leading to reduced congestion, improved information transfer efficiency, and heightened service quality within the network.

The remainder of this paper is structured as follows. Section II presents a background of the problem. Section III provides a detailed description of the proposed method, outlining the key steps and innovative aspects of our approach. In Section IV, we present the results of empirical evaluations and comparisons, demonstrating the superiority of our method in terms of accuracy and efficiency. Finally, Section V summarizes our findings, highlights the contributions, and suggests potential avenues for future research.

II. Background

Performance parameters within a network are difficult to measure directly by internal nodes as some may not communicate proactively for security reasons. The network tomography method provides a very convenient means of measuring network parameters since it measures only the end-to-end characteristics of the network. In the end, the network is estimated based on its internal features. Many approaches are employed in network tomographies, such as estimating link level 1 parameters, determining network correlations, and calculating the network traffic matrix between the source and destination networks.

Several methods are presented for estimating link delay via a tomography network. In some cases, the delay distribution is discrete. These methods assign end-to-end latencies and delay links based on assumed collections, and the probability of mass-delay links is calculated using the EM algorithm [25]. In a statistical model containing latent dependent variables that are imperceptible, the EM algorithm or maximum expected (expectation-maximization) is a repeatable method for finding the maximum value or estimate of the most likely value of an inductive (inferring from effects to causes) parameter set of parameters. The calculation will be repeated for most parameters in the expected results. Calculations and estimates are obtained in stage M to determine the distribution of latent variables used in the next iteration (E) [26]. The method used to estimate the delay of end-to-end latencies supposed links to collections are assigned. The possibility of mass-delay links is calculated using the EM algorithm. There are two limitations in the face of real networks:

- In a real network, because the traffic load on some of the links is heavier and lighter for the other part of the link, packet delay can be quite different. So, if you set that delays end-to-end, links have been assigned to the minor premise. In this case, the desired levels of accuracy are not achieved. Suppose these sets are assumed to be large. In that case, complex computational problems arise, so selecting the appropriate set to assign end-to-end latency and link latency is impossible for these methods.
- The large network size, usually with large amounts of complex data (sets assumed that the end-to-end delays and delays links are assigned to them) to obtain delays of all links face. However, the EM algorithm calculates the probability of each of the probe packets. However, if the input data set is large, these are time-consuming calculations and thus cannot link estimates on time delays for large-scale networks [27].

Unlike the discrete distribution models that estimate the delay, some methods are used to model continuous distribution delay. These methods assume that the distribution of the known delay follows. Examples of distributions include the Poisson distribution and Gaussian distribution compound [26, 28]. Peterson and Davie [29] The authors sought to improve the algorithm to Kaczmarz to take advantage of computing linear tomography systems in the network. The algorithm in linear systems, adaptive control, and tomography systems with wireless sensor networks is used in nature and later as an iterative algorithm in these cases. This algorithm is based on mathematics, potential events, and the full and complex.

In [30], routing and tomography have been amended using a series of evolutionary and biological matrices added and the idea of using algorithms. The framework of evolutionary algorithms solving the robustness and accuracy helps tomography. More studies in the area of network tomography are based on parametric models. The parametric model assumes that measuring data traffic depends on the number of defined parameters. For example, recent studies estimate the internal delay distribution. The probability mass distribution can be modeled as functions. In this context, parameters and the probabilities associated with each function are likely crimes. Some methods of inference delay focused on multicast routing. The routing of packets sent from the sender to the recipient during an operation moved. During the probe packets, the paths split apart, double, and multiply [31].

III. Proposed Method

With the advent of 5G and the impending rise of 6G, the networking landscape has been marked by transformative capabilities, promising higher data rates, lower latency, and greater network efficiency. These advancements are largely attributed to the proliferation of advanced wireless technologies, the expansion of device connectivity, and the integration of cutting-edge communication paradigms. As the industry shifts its focus toward ultra-low latency, these technologies are poised to reshape the dynamics of data transmission and drive the development of novel applications. In this dynamic context, the accurate estimation of Round-Trip Time (RTT) assumes heightened significance. RTT estimation plays a pivotal role in determining optimal retransmission timeouts, facilitating congestion control, and enhancing overall network performance. While conventional methods have sufficed in earlier network paradigms, the evolving landscape introduces new challenges and opportunities that necessitate innovative approaches.

In order to maximize network efficiency, accurate RTT estimation and optimal RTO quantification are essential. The RTO must be longer than the return time or RTT. The RTT is affected by many factors, such as transmission delay, propagation delay, header processing time, ACK production time, etc. Consequently, RTT is not a static value in real-world environments and will change over time. In this case, the change reflects the conditions mentioned in the examples. The RTO should not be set with more than enough, as this will lead to long delays. Additionally, there is an important question to be addressed: what happens if the route changes? What should be done if the network traffic situation and the status of the intermediate nodes change? It is, therefore, necessary to repeat this estimate regularly. The period can be repeated with an appropriate interval of time, such as 20 milliseconds. In order to send probes, a condition may be set, which will result in re-sending the packets if that condition is met. Network conditions will be estimated correctly again. Knowledge of any link to update, to help without sending the actual data and only based on the information provided by the source node is placed tomography, the parameters needed to set up protocols like TCP.

These parameters may also be estimated based on other ideas for calculating them. By estimating the RTT more accurately, the TCP can retransmit at better times and increase the number of these posts. In order to overcome the problems mentioned above, a method based on the unicast delay estimates tomography is recommended for improving TCP. Tomography is used in this way for both end-to-end and link-to-link tomography. Tomography can be used by any network node to calculate the delay on links, whether it is the transmitter or any other node on the network.

Furthermore, it can assist in improving the routing process. To estimate network delay, we first use tomography. To estimate the number of packet delay probe pairs, unreachable packets are sent from the source node to the destination node. End-to-end packets are bursts that act as end-to-end communications. A delay equation is developed using the relationship between the delay and the delay link that routes packets of information to the probe. Our solution is derived from equations described in the previous chapter to estimate the quantity delay optimally. The route is calculated based on the number of nodes, and each node will have some delay in packets. In general, course delays are little more than the sum of the delays from link to link. As each node experiences a delay in processing, queuing, or propagation, connections are also created. The decision to send real-time packets may result in incorrect calculations and the loss of influence tomography performed to adjust the network RTT in the TCP protocol when the decision is made to send packets in real-time.

In spite of the fact that the processing delay may seem insignificant at first, it does not have any significant impact on large networks with high traffic, especially when there are a number of intermediate nodes and those with high and low processing power. Each node is given a delay. Additionally, it is also possible to have a significant other since it is always in the real world, with a variety of unforeseen and additional delays. Besides the low estimate of the time required for RTT retransmission, a high estimate of the real-time network and reach pass may help reduce the package's delay. When the estimate for the time is slightly larger, it is better to be low since a low estimate, increased network traffic, and poor network conditions can all contribute to delays. By sending and receiving information, each probe packet can help estimate the exact real delay.

For closed questions, the collection, analysis, and length of time from the time of receiving the low are examined, and the amount of delay is determined based on the link. A more appropriate RTT parameter is determined based on the level of probe tomography accuracy of information and links to the detailed estimate of the actual delay. The RTT value is calculated using probes that measure the RTT by tomography and real delay roadshows, achieving a much more accurate delay than TCP. Finally, we will use the results of this calculation to regulate better and improve our TCP retransmission mechanism. By using the proposed approach, better estimates of RTT TCP can be made, and thus, better performance may be achieved.

A. *Approach Proposed Resolution*

As RTT is the basis for accurately estimating probe tomography, multiple packets of burst tomography can be sent to different recipients using unicast protocols. These probes will be used to determine the answer depending on the sender and RTT. The following steps are followed to estimate the actual RTT.

In the first step, tree topology is considered for the network. Leaf nodes of the tree network are selected as the destination. To send pairs of probe packets, multi-packet probe packets are burst using unicast routing. Each packet transmits information about each link, including when it was sent and received by each node, enabling a more accurate estimate of the latency of the links and the route. Data collection and analysis of transmitter probe packets and the amount of time calculated by the probe packet are low. The amount of packet delay is calculated according to the network. During the extraction and analysis process, we provide accurate, real-time probe data packet processing at each node to estimate the actual RTT at each node. The information obtained from the probe packets is used to calculate the actual delay based on the optimal number of links as follows:

Path Delay = (Tomo Delay) + ((2n-2) * (Mean Node Delay)) (1)

In Eq. (1), Tomo Delay is the optimal amount of delay quantity for links through tomography, n is the number of nodes between source and destination, and mean Node Delay represents the average delay of processing nodes. The probability does not decrease if the sweep is different and has a different number of nodes. If there were more nodes along the path, the equation was recalculated. By replacing parameter 2n with the number of nodes and the number of nodes in the path, the equation is modified as follows:

Path Delay = (Tomo Delay) + ((n_went + n_back) * (Mean Node Delay)) (2)

To improve the mechanism of retransmission of TCP, the result calculation shows that the actual delay replaces the RTT value calculated by the TCP protocol so that TCP will begin with a more accurate estimate of RTT. In practice, more complete equations may be available instead of replacing the equation. There is, however, an additional condition that can be met by a simple equation to estimate the actual delay. The routine can be effective for a network. A better match will always be required in computational experiments.

B. *Evaluation of the precision of the delay*

This section provides information regarding the delay time, accuracy, and reliability. In addition to accurately representing scientific work, it indicates what they can contribute to other research endeavors. Estimated delay in TCP is considered an important issue, and its accuracy is crucial. We use the cumulative method to adjust and improve TCP RTT. If the estimates are correct, they will have a significant impact on the performance of the network. Statistical analysis indicates that the proposed method effectively predicts the delay, and the results indicated they would be considerable. It is, therefore, necessary to compare the delay used to calculate the cumulative delay in the proposed method with the actual delay by the time an estimate is made regarding the links and integrated.

The complexity and variability of the network make it impossible to test with a definitive result. So, the effort involved here has been estimated at ten times. This means that every time, a predetermined scenario is applied to every package in exchange for the time a probe is sent, calculated, printed, and recorded at the outlet. Additionally, a comparison was made between the sent and received packets obtained and recorded, simultaneously with the actual delay time on the printed output. Because the two packages are sent on two separate paths, each package will experience two delays. Therefore, the two delays are calculated every time. The third scenario is somewhat busier in the network regarding the calculation. The traffic in this scenario consists of 21 TCP streams and 161 on-off Poisson streams transmitted over UDP. Statistics are collected from node one. It can be said that the accuracy of this method is relatively high, and errors in the tenth milliseconds. For more detailed calculations, each row of accuracy and the error rate can be calculated and then averaged. The packet error rate refers to the number of packets with errors divided by the number of packets transmitted, calculated by Eq. (3).

$$PER = \frac{number\ of\ packet\ with\ errors}{number\ of\ packets\ transmitted} \tag{3}$$

Table I provides a comparison of the actual delay and delay calculation for two batches of the probe. This data illustrates the accuracy of our proposed method in estimating delays accurately. Similarly, Table II presents the percentage of delay calculation error in the proposed method for two packs of probe data. Analyzing this data allows us to assess the precision of our approach in real-world scenarios.

TABLE I. COMPARISON OF ACTUAL DELAY AND DELAY CALCULATION FOR TWO BATCHES OF THE PROBE

Closed computational delay the first	**Real delay packet first**	**Delay calculation package II**	**Real delay packet second**
42/5 ms	23.5 ms	10.6 ms	18.6 ms
34.6 ms	87/6 ms	32.7 ms	92/6 ms
5.55 ms	90/5 ms	21/6 ms	7.17 ms
89.7 ms	11.8 ms	29/4 ms	80.4 ms
24/8 ms	91/7 ms	81.7 ms	63/7 ms
94/4 ms	18.5 ms	23.7 ms	65/7 ms
34.5 ms	49/5 ms	48/6 ms	11.7 ms
87.5 ms	08/6 ms	76/3 ms	45/3 ms
12.6 ms	67/6 ms	22/8 ms	59/7 ms
34.8 ms	79/7 ms	34/7 ms	8.57 ms

TABLE II. PERCENTAGE OF A DELAY CALCULATION ERROR IN THE PROPOSED METHOD FOR TWO PACKS OF PROBE

Percent packet error First	Packet error percentage of the second
60/3 percent	20.1 percent
70/7 percent	70/5 percent
90/5 percent	30/13 percent
71/2 percent	80.4 percent
60/10 percent	30.2 percent
60/4 percent	40/5 percent
70/2 percent	80/8 percent
40/3 percent	90/8 percent
20.8 percent	30.8 percent
00/7 percent	40.6 percent

C. The potential of network tomography-based RTT estimation

In this backdrop, our proposed method offers a novel approach to RTT estimation, leveraging the power of network tomography. Traditional RTT estimation methods may confront limitations in accurately gauging RTT in scenarios with multiple physical paths, varying link delays, and changing network dynamics. Here, network tomography emerges as a promising avenue to address these challenges by providing end-to-end performance insights based on the measurement of delay characteristics. Our method, grounded in network tomography, introduces an additional dimension to RTT estimation. While advancements in networking technologies, such as 5G and 6G, and increased compute power contribute to RTT reduction, our method augments these efforts by enhancing accuracy and reliability. By considering link delays across multiple paths and leveraging network tomography's capabilities, our method provides a more comprehensive and nuanced RTT estimation. Fig. 1 shows the flowchart for actual RTT estimation.

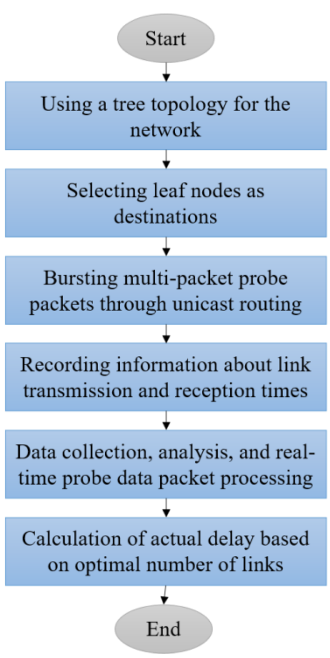

Fig. 1. RTT estimation.

D. Relevance in evolving networking technologies

The evolving networking technologies, particularly 5G and the anticipated 6G, demand precision in RTT estimation to harness their full potential. While these technologies offer low-latency communication, our method's emphasis on accurate RTT estimation aligns with the quest for precision in data transmission. In environments where real-time applications, IoT devices, and mission-critical communications are paramount, our method's ability to adapt and estimate link delays across diverse paths becomes indispensable. In scenarios with multiple physical paths, MPTCP utilization, and the intricate interplay of network conditions, our method demonstrates its relevance by providing insights into delay characteristics that traditional methods may overlook. This nuanced estimation approach aligns well with the objectives of evolving networking technologies to minimize latency and optimize data transfer efficiency.

IV. EXPERIMENTAL RESULTS

Simulation may take into account particular circumstances. Simulating and evaluating all possible scenarios is neither feasible nor logical. In each simulation, the actual network conditions should be considered, but the results should be reported as accurately as possible under the constraints. Furthermore, many scenarios are proposed for simulating various aspects of the proposed method. Simulations must also be conducted under conditions similar to those encountered in the review, compared with the method before using it or other suggested methods. In some contexts, a comprehensive network may not be necessary and may be subject to performance conditions. The algorithm is not random in nature, and its behavior is uncertain. Moreover, while improving the network from the perspective of all available parameters, other parameters may be affected. This is because certain reconciliation parameters, namely the balance between them, as well as improving all the parameters simultaneously, cannot be achieved.

The NS2 simulator is used to simulate the proposed method. In order to implement several networks, OTcl is used to create scenarios, and C++ code is used to simulate algorithms and protocols. Communication between scripting languages should be considered in this regard. Although the NS2 software is available on Linux, it can be used on Windows via a software interface called Cygwin, a Linux virtual machine that can run NS2. The proposed method is simulated and evaluated on a wired network at a fixed time. At this time, seven of eight nodes are linked to a tree with a root node number of zero. Fig. 2 shows this package.

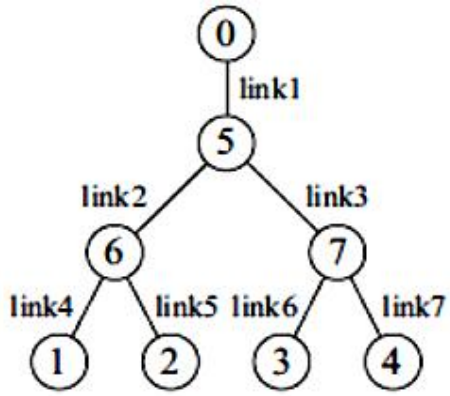

Fig. 2. A network of assessment.

Each link has a queue that is a FIFO algorithm. The simulation was performed with three types of traffic. Traffic is the most crucial first tomography is used, TCP traffic that RTT adjustment is considered. Two types of background traffic are assumed as the frequency of use. CBR traffic, Poisson traffic, and others with entirely different features. Poisson traffic is a traffic accident, while CBR traffic congestion is at the rate specified and is absolutely certain. The CBR traffic intended as a background other than CBR traffic that was initially closed probe is used to estimate RTT TCP traffic. The proposed and other methods and evaluations were conducted in two different scenarios.

A. Network Topology

The experimental setup involves a wired network comprising nodes connected in a tree topology, with a root node numbered zero. The nodes are connected through links equipped with a First-In-First-Out (FIFO) queueing algorithm. Two distinct scenarios are considered, each reflecting different aspects of network dynamics and traffic patterns.

Scenario 1 involves a multi-traffic flow setting, where TCP flows are established at rates of 21 units. These TCP flows coexist with background traffic consisting of CBR traffic and Poisson traffic, which is modeled using the UDP. This scenario creates a dynamic and diverse network environment by combining TCP flows with different rates and various types of background traffic, such as CBR and Poisson traffic. The primary objective of this scenario is to thoroughly evaluate and assess the adaptability of the proposed method under conditions of varying network loads and diverse traffic types. Scenario 2 explores the interplay between CBR and TCP dynamics, where TCP flows operating at rates of 21 are integrated with both CBR traffic and background Poisson traffic. The emphasis transitions towards evaluating the effectiveness of the suggested approach when contending with clearly defined CBR traffic, contributing a heightened level of foreseeability to the network ambiance. This particular setting furnishes valuable observations regarding the method's proficiency in situations characterized by more predictable traffic arrangements. A tree topology governs data routing within the simulated network. The rationale behind this choice lies in its ability to capture a simplified representation of network structures, enabling controlled experimentation while providing a foundation for performance evaluation.

B. Results and Analysis

The proposed method's performance is evaluated under a network load with a mix of TCP flows, CBR traffic, and Poisson traffic in the first scenario. The goal is to assess how the algorithm adapts RTT adjustments to optimize network performance under diverse conditions. Fig. 3 and 4 illustrate the comparison of throughput and delay across multiple simulations. The results highlight the method's ability to maintain efficient data transfer despite varying traffic types and network load. The second scenario delves into the method's interaction with competing CBR traffic and effectiveness in a more deterministic setting. Fig. 5 and 6 show the throughput and delay comparisons. The method demonstrates its capability to achieve competitive performance under such conditions, albeit with varying degrees of effectiveness compared to scenario 1. The proposed method's performance is evaluated against different traffic types and rates in both scenarios. The nuanced results reflect the method's adaptability to diverse network conditions and the potential to optimize RTT adjustments effectively. The results underscore the method's potential to provide accurate RTT estimations and optimize data transmission despite varying network dynamics. As networking technologies evolve, our method's capacity to enhance RTT estimation remains relevant and valuable, contributing to optimizing data transfer mechanisms in the face of complex and diverse networking scenarios.

Our proposed method for RTT estimation through network tomography offers versatile, practical applications across diverse contexts. In real-time communication systems, such as telemedicine and remote surgery, where low latency is critical for seamless interactions, our method ensures accurate RTT estimation even in complex, multi-path networks. In IoT deployments, where devices often rely on efficient data exchange for timely decision-making, our approach enhances network reliability by precisely estimating RTT, leading to optimized device communication. Furthermore, in vehicular networks, where vehicles exchange safety-critical information, our method's adaptability to changing network conditions ensures reliable RTT estimation, contributing to safer road environments. In cloud computing environments, where data transfer efficiency impacts application performance, our method aids in precise RTT estimation, leading to enhanced user experiences. These use cases highlight the tangible benefits of our proposed scheme, underscoring its potential to revolutionize diverse industries by improving RTT estimation accuracy, network reliability, and overall performance.

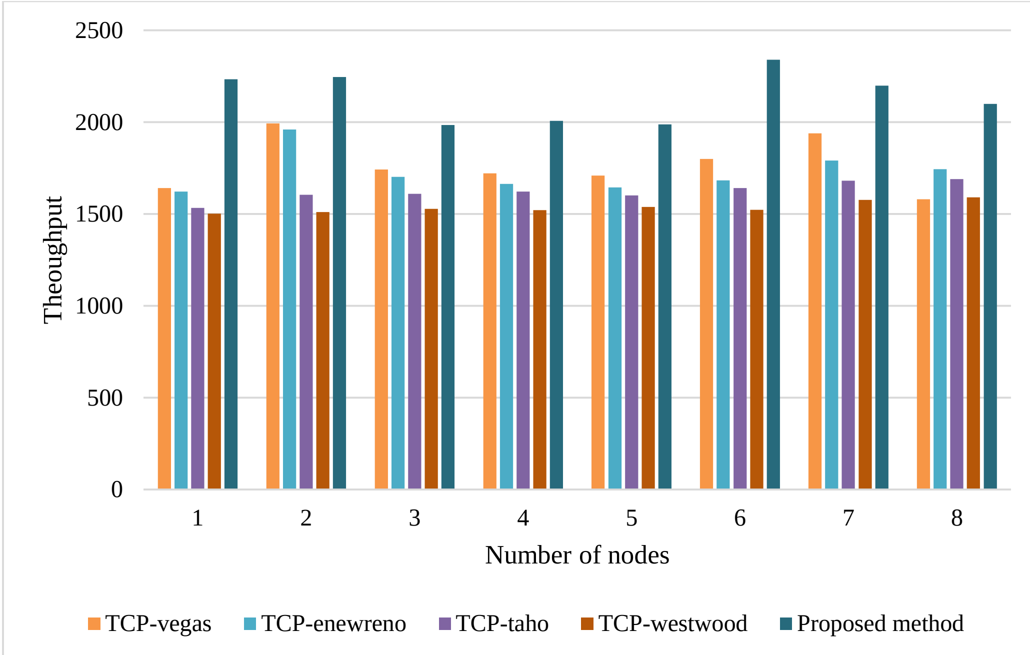

Fig. 3. Comparison of the throughput in the first scenario.

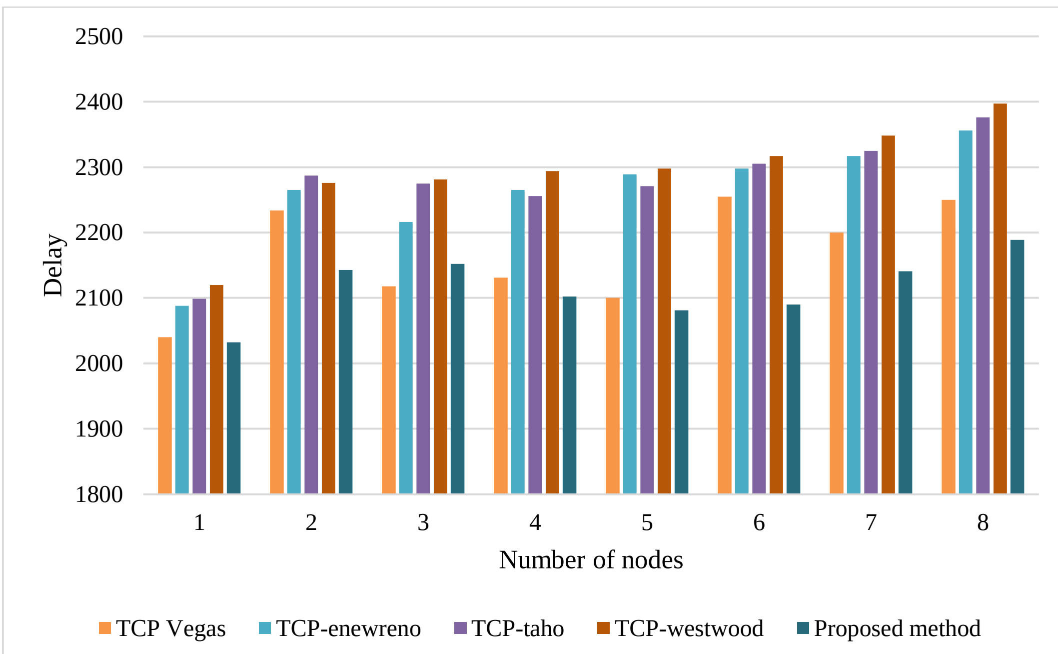

Fig. 4. Comparison of the delay in the first scenario.

While our proposed method significantly enhances TCP's RTT estimation, it is essential to acknowledge that our approach primarily focuses on improving RTT accuracy and reducing congestion in the absence of packet corruption. In cases where packet corruption occurs during transmission, the TCP protocol is equipped with error detection mechanisms, such as checksums, to identify and request retransmission of corrupted packets. Our work primarily complements these existing error detection and recovery mechanisms by providing more precise RTT estimations, which can lead to more efficient congestion control. However, in cases of severe packet corruption or loss, additional mechanisms at higher protocol layers, such as transport layer error correction codes or application-layer retransmission strategies, may be necessary to ensure data integrity and reliability.

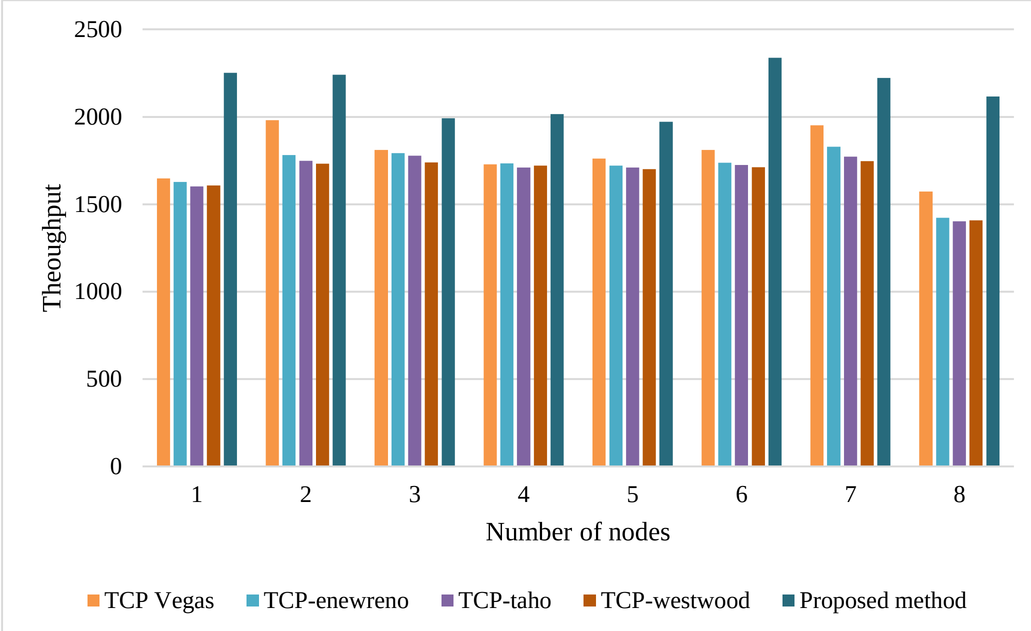

Fig. 5. Comparison of the throughput in the second scenario.

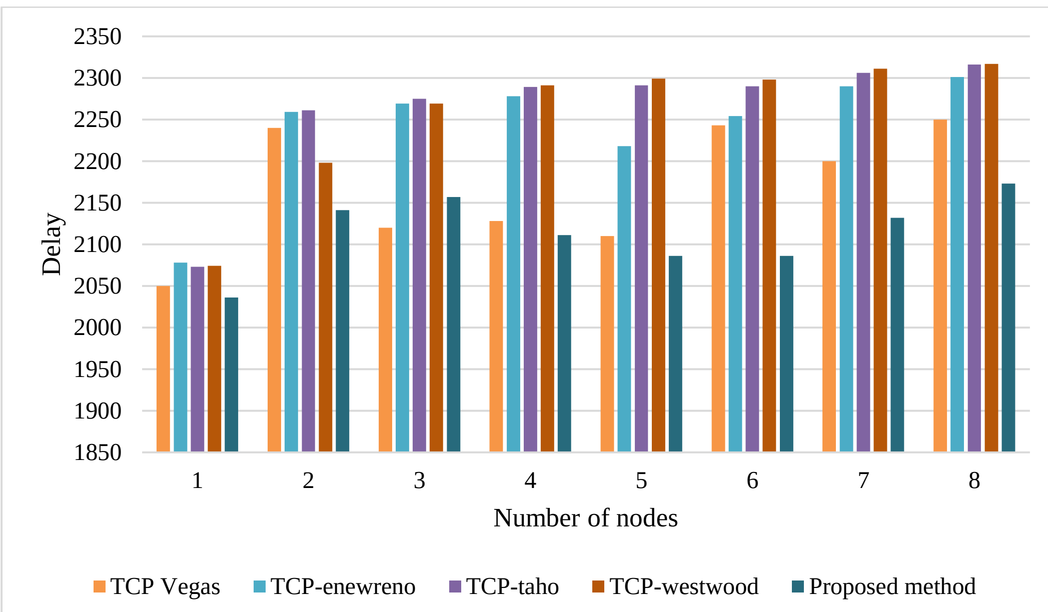

Fig. 6. Comparison of the delay in the second scenario.

V. Conclusion

Network tomography offers an invaluable method for analyzing a network's performance. To determine internal performance parameters, network tomography measures end-to-end performance, as opposed to methods that are based on internal communication. This method has the capability of estimating times in the event that it is necessary to evaluate a protocol's performance in a network by estimating times. The TCP protocol can be significantly improved by properly estimating the RTT time. The proposed method is a relatively flexible method that can be performed using different settings and conditions, and other formulas receive different results. The proposed method is a valuable mechanism for different computer networks to estimate the rate of actual delay, which deals with network conditions and restrictions to ensure the quality of their service is targeted.

While our method significantly improves RTT estimation, it may not achieve absolute accuracy due to factors such as network variability and unforeseen delays. The sensitivity of our approach to rapid network dynamics necessitates periodic RTT updates, but it may not capture instantaneous changes. Moreover, computational complexities and resource requirements should be considered in resource-constrained environments. Real-world networking scenarios can introduce challenges beyond modeling capabilities, and the scope of our empirical evaluations may not cover all possible conditions. These limitations provide valuable context for the application and interpretation of our research findings.

ACKNOWLEDGMENT

This research is funded by the Henan Province Science and Technology Project (grant No.232102210117,192102210285, 152102210023) and the Research Project of Higher Education Teaching Reform of Huanghuai University (grant No. 2021XJGLX51), and Henan Engineering Research Center of Big Data Analysis and Application for Equipment Manufacturing Internet of Things.

REFERENCES

[1] J. Khan et al., "SMSH: Secure surveillance mechanism on smart healthcare IoT system with probabilistic image encryption," IEEE Access, vol. 8, pp. 15747-15767, 2020.

[2] J. Khan et al., "Efficient secure surveillance on smart healthcare IoT system through cosine-transform encryption," Journal of Intelligent & Fuzzy Systems, vol. 40, no. 1, pp. 1417-1442, 2021.

[3] C. Casetti, "5G Standalone Deployments Are on the Rise [Mobile Radio]," IEEE Vehicular Technology Magazine, vol. 18, no. 1, pp. 5-11, 2023.

[4] A. J. Pinheiro, J. d. M. Bezerra, C. A. Burgardt, and D. R. Campelo, "Identifying IoT devices and events based on packet length from encrypted traffic," Computer Communications, vol. 144, pp. 8-17, 2019.

[5] P. Tomar et al., "Cmt-sctp and mptcp multi-path transport protocols: A comprehensive review," Electronics, vol. 11, no. 15, p. 2384, 2022.

[6] G.-M. Sung, C.-T. Lee, Z.-Y. Yan, and C.-P. Yu, "Ethernet Packet to USB Data Transfer Bridge ASIC with Modbus Transmission Control Protocol Based on FPGA Development Kit," Electronics, vol. 11, no. 20, p. 3269, 2022.

[7] L. Zong, H. Wang, and G. Luo, "Transmission Control Over Satellite Network for Marine Environmental Monitoring System," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, pp. 19668-19675, 2022.

[8] H. Bennouri and A. Berqia, "U-NewReno transmission control protocol to improve TCP performance in Underwater Wireless Sensors Networks," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 8, pp. 5746-5758, 2022.

[9] M. Sarbaz, I. Zamani, M. Manthouri, and A. Ibeas, "Hierarchical optimization-based model predictive control for a class of discrete fuzzy large-scale systems considering time-varying delays and disturbances," International Journal of Fuzzy Systems, vol. 24, no. 4, pp. 2107-2130, 2022.

[10] M. Sarbaz, M. Manthouri, and I. Zamani, "LMI-Based Robust Fuzzy Model Predictive Control of Discrete-Time Fuzzy Takagi-Sugeno Large-Scale Systems Based on Hierarchical Optimization and H∞ Performance," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 30, no. 04, pp. 649-679, 2022.

[11] T. Gera, J. Singh, A. Mehbodniya, J. L. Webber, M. Shabaz, and D. Thakur, "Dominant feature selection and machine learning-based hybrid approach to analyze android ransomware," Security and Communication Networks, vol. 2021, pp. 1-22, 2021.

[12] R. Singh et al., "Analysis of Network Slicing for Management of 5G Networks Using Machine Learning Techniques," Wireless Communications and Mobile Computing, vol. 2022, 2022.

[13] J. Webber, A. Mehbodniya, Y. Hou, K. Yano, and T. Kumagai, "Study on idle slot availability prediction for WLAN using a probabilistic neural network," in 2017 23rd Asia-Pacific Conference on Communications (APCC), 2017: IEEE, pp. 1-6.

[14] T. Taami, S. Azizi, and R. Yarinezhad, "Unequal sized cells based on cross shapes for data collection in green Internet of Things (IoT) networks," Wireless Networks, pp. 1-18, 2023.

[15] M. Bagheri et al., "Data conditioning and forecasting methodology using machine learning on production data for a well pad," in Offshore Technology Conference, 2020: OTC, p. D031S037R002.

[16] R. Soleimani and E. Lobaton, "Enhancing Inference on Physiological and Kinematic Periodic Signals via Phase-Based Interpretability and Multi-Task Learning," Information, vol. 13, no. 7, p. 326, 2022.

[17] M. Shahin et al., "Cluster-based association rule mining for an intersection accident dataset," in 2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube), 2021: IEEE, pp. 1-6, doi: 10.1109/ICECube53880.2021.9628206.

[18] S. Saeidi, S. Enjedani, E. Alvandi Behineh, K. Tehranian, and S. Jazayerifar, "Factors Affecting Public Transportation Use during Pandemic: An Integrated Approach of Technology Acceptance Model and Theory of Planned Behavior," Tehnički glasnik, vol. 18, pp. 1-12, 09/01 2023, doi: 10.31803/tg-20230601145322.

[19] U. Jamil, M. Sulaiman, N. Ghafoor, M. Malmir, F. Nawaz, and R. I. Shakoor, "Power Harvesting towards Sustainable Energy Technology through Ambient Vibrations and Capacitive Transducers," in 2023 International Conference on Emerging Power Technologies (ICEPT), 2023: IEEE, pp. 1-6.

[20] G. Amponis, T. Lagkas, K. Tsiknas, P. Radoglou-Grammatikis, and P. Sarigiannidis, "Introducing a New TCP Variant for UAV networks following comparative simulations," Simulation Modelling Practice and Theory, vol. 123, p. 102708, 2023.

[21] C. Lim, "Improving congestion control of TCP for constrained IoT networks," Sensors, vol. 20, no. 17, p. 4774, 2020.

[22] M. Nikzad, K. Jamshidi, A. Bohlooli, and F. M. Faqiry, "An accurate retransmission timeout estimator for content-centric networking based on the Jacobson algorithm," Digital Communications and Networks, vol. 8, no. 6, pp. 1085-1093, 2022.

[23] R. Tagyo, D. Ikegami, and R. Kawahara, "Network tomography using routing probability for undeterministic routing," IEICE Transactions on Communications, vol. 104, no. 7, pp. 837-848, 2021.

[24] A. Ibraheem, Z. Sheng, G. Parisis, and D. Tian, "Neural network based partial tomography for in-vehicle network monitoring," in 2021 IEEE International Conference on Communications Workshops (ICC Workshops), 2021: IEEE, pp. 1-6.

[25] L. Breuer and A. Kume, "An EM Algorithm for Markovian Arrival Processes Observed at Discrete Times," in MMB/DFT, 2010: Springer, pp. 242-258.

[26] G. Antichi, A. Di Pietro, D. Ficara, S. Giordano, G. Procissi, and F. Vitucci, "End-to-end inference of link level queueing delay statistics," in GLOBECOM 2009-2009 IEEE Global Telecommunications Conference, 2009: IEEE, pp. 1-6.

[27] G. Fei, G. Hu, and X. Jiang, "Unicast-based inference of network link delay statistics," in 2011 IEEE 13th International Conference on Communication Technology, 2011: IEEE, pp. 146-150.

[28] M.-F. Shih and A. Hero, "Unicast inference of network link delay distributions from edge measurements," in 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221), 2001, vol. 6: IEEE, pp. 3421-3424.

[29] L. L. Peterson and B. S. Davie, Computer networks: a systems approach. Elsevier, 2007.

[30] R. M. Castro, M. J. Coates, and R. D. Nowak, "Likelihood based hierarchical clustering," IEEE Transactions on signal processing, vol. 52, no. 8, pp. 2308-2321, 2004.

[31] J. Lin, J. Zhang, and W. Lin, "The improved method of moment for multicast-based delay distribution inference in network tomography," in 2010 2nd International Conference on Advanced Computer Control, 2010, vol. 4: IEEE, pp. 486-489.