Iraqi Journal of Intelligent Computing and Informatics (IJICI)
Vol. 1, 2, December 2022, pp. 119~130
ISSN: 2791-2868, DOI: 10.52940/ijici.v1i2.22

Enhanced system for ns2 trace file analysis with network performance evaluation

Jawad Alkenani[1*], Khulood Ahmed Nassar[2]
[1]Department of Computer Science, Shatt Al-Arab University College, Basra, Iraq
[2]College of Computer Science and Information Technology, Computer Information Systems Department, University of Basrah, Basrah, Iraq

Article Info

Article history:

Received November 20, 2022
Revised December 12, 2022
Accepted December 21, 2022

Keywords:

Language Java
Network simulator
Network performance
Quality of service
Trace graph

ABSTRACT

One of the critical challenges facing operators around the world is how to ensure that everything is running smoothly as well as how to analyze the performance of the network. Nonetheless, the analytic system must be precise, user-friendly, and quick enough to depict network performance in real time. Network performance is essential for ensuring service quality. In this light, the Network Simulator NS-2 is generally employed for network research on the widely-used UNIX and Windows systems. Next, the network scenarios are generated using network simulation scripts, and upon completion of the simulation, trace files that record network events are generated. Information that might be utilized in a performance study could be recorded in trace files. This could include the total number of packets sent, their arrival times, any delays or losses, and so on. This trace file analysis problem arises mostly because NS-2 does not provide any visualization options for analyzing simulation results (trace files). A system is proposed to analyze the NS2 trace file and facilitate data extraction and evaluation. The main feature of this system is the Java language for network performance evaluation. And to prove the validity of the system, the system was compared with the Trace graph, where the speed of the new system was very high.

Corresponding Author:

Jawad Alkenani
Department of Computer Science, Shatt Al-Arab University College, Basra, Iraq
Email: Jawadalkenani@sa-uc.edu.iq

1. INTRODUCTION

Complex monitoring techniques are required in computer networks to keep track of key performance indicators like latency and packet loss in real time. These next-generation monitoring systems need to not only quickly detect network performance deterioration but also identify the root cause of service quality problems to fulfill the stringent standards for voice and video services. Thus, these monitoring tools will be crucial in guaranteeing service quality and avoiding service interruptions by allowing fast resolution of network issues (such as rerouting traffic around a busy connection). On the other hand, passive monitoring approaches extrapolate network performance by listening in on data passing over network links. Passive monitoring offers many advantages over active monitoring [1], [2]. First, it does not contribute monitoring-related synthetic traffic to the network architecture. This is especially crucial when a network interface or connection gets congested since injecting more traffic for active measurements would just worsen the issue. Second, because all measurements are based on real network traffic, passive approaches may accurately examine network performance as it is seen by end users. Consequently, the Network Simulator NS-2 is mostly used for network research [3]. The reason behind its use is that it is spread all over the world, especially in the past ten years, computer networks have developed significantly, prominently on Internet, networks are growing more than in the past and have become more complex. It supports different workstations leading to faster transmission than before and improving connection capabilities. The necessity for tools to monitor and analyze network communications was necessitated by the continuous growth and increasing functionality needs of networks[1]. It resulted in the development of network simulators, which help to design and analyze different types of networks, protocols, etc. NS-2 simulator is developed at Laurence Berkeley National Laboratory (LBNL). Nowadays, 44% of researchers use NS-2 simulators because NS-2 can support the user to add its modules as well as low cost of development and implementation in comparison to experimental tests in real environments [4]. This research contributes by demonstrating how an NS-2 trace file, after being parsed and displayed, may be viewed rapidly, simply, and effectively. Using the NS-2 network simulation environment, the Java-based NS-2 network (JDNA) analyzer is described in this project as a tool for collecting, analyzing, and visualizing trace data. Therefore, the user would be able to assess network performance through an interactive graphical user interface (GUI), saving them time over manually processing data using script tools. The main problem with this trace file analysis challenge is that NS-2 does not provide any visualization options for analyzing simulation results (trace files)[5]. Various NS-2 trace analyzers are proposed to facilitate the extraction of data for performance evaluation. The key characteristic of these works is the language Java of existing service quality[6], [7].

This paper delves into the process of developing a cheap passive monitoring system for determining a metric of network performance. The suggested system is a Java-based program that reads Network Simulator 2 (NS-2) files and pulls out relevant network events. After detecting and analyzing trace files, the system presents the information contained inside a GUI together with other data, metrics, and statistics linked to the simulation.

This essay is structured as follows: In Section 2, past work on network performance analysis is given. The third section describes the measures used to evaluate network performance. The fourth section discusses a suggested Java tool for analyzing Ns2 trace data. The conclusion is stated in Section 5 of this work.

2. RELATED WORK

A number of experts have noted that a new method for quality monitoring and network performance assessment emerged in 2021[8]. Device-to-device (D2D) interactions enhance cellular network capacity. D2D allows mobile users to communicate directly without going through the base station (BS), however, it can interfere with cellular networks, reducing system performance. To ensure effective and reliable D2D operations within a cellular network, the design of a system containing both D2D and cellular users must satisfy QoS concerns. In this research, they evaluated three alternative transmission techniques (cellular, D2D, and cooperative D2D modes) by generating closed-form expressions of QoS performance such as system dependability, viable data rate, and energy efficiency. Adaptively picking transmission schemes and favorable system trade-offs can maximize the proposed strategy, and the objective outlined by the authors in [9]is to enhance patient care by minimizing latency and packet loss. The network measurement was computed using Transmission Control Protocol's Congestion Window (TCP). Based on the trace file, which is the output of the simulator, all parameters are analyzed. However, the proposed research does not specify the network format supported, and external tools are used to generate the statistical data. In the year 2020 authors [10], Examine the effectiveness of various routing strategies for mobile ad hoc networks using a range of metrics. The evolution's performance varies across four critical metrics, Increases in the number of mobile nodes and the packet sizes may cause a variety of different metrics to shift. However, the authors perform all the experiments using script language, which makes the assessment procedure more time-consuming when compared to other script languages. The authors in [11], Real-Time Networks (RTNs) provide latency assurances for time-sensitive applications and strive to accommodate diverse traffic categories through various scheduling techniques. These scheduling algorithms rely on an accurate measurement of network performance to dynamically alter the scheduling strategies. Machine Learning (ML) provides a recursive method for measuring network performance. Network Calculus (NC) is capable of calculating the boundaries for the primary performance indices, including latencies and throughputs, in an RTN for ML. Thus, the merging of ML with NC enhances the overall efficiency of calculations. The authors in [12], Examine how well mobile IP works on ad hoc networks. The authors, like those in similar earlier studies, spent extra time using an external programming language to do the analyses. An effective performance assessment tool is one with an attractive graphical user interface that makes the procedure simple. Present the authors in 2015[13], there is a lack of network measurement, and the data and graphical reports given are limited. Despite this, the authors provided a novel program that could examine the trace file and generate several statistical data. However, several aspects are inadequate and need improvements, such as the suggested tool's incompatibility with all operating systems and inability to handle multi-file formats. So, presented to the authors in 2015[14], This study used many graphical interfaces to assess the simulator's output. However, it cannot read multiple trace files concurrently and cannot analyze all trace file kinds. present the author in this study[15], the interface tool was capable of displaying many reports and statistical findings.

Researchers can examine network performance to determine if it is satisfactory or not. However, while the program can display a number of metrics, it can only read two trace files. Video streaming is difficult since it is sensitive to changes in transmission quality, as demonstrated by the Researchers in [16]. Being able to put a number on the quality of service received is useful in streaming apps. In order to improve the perceived playback quality, data from QoS measurements can be used to adapt video traffic to the network's transmission constraints. It is the goal of this project to examine the concept of quality of service, look at several quality of service monitoring approaches, and create a system that monitors the quality of service from beginning to end for multiple concurrent video streaming sessions. The study's methodology is based on a constructive analysis of pertinent literature and technology, and its findings come from an end-to-end quality-of-service monitoring system. present the author of this study[16], a tool capable of analyzing network performance and revealing the quality of the networks was introduced. Nevertheless, not all network formats are supported by this work. Additionally, the tool can only process a single trace file, making the analysis of different configurations more difficult. There are active and passive probes presented in [18] that help service providers spot problems in service. We calculate the network metrics (delay, loss, and jitter) that may be utilized to calculate the call quality with a voice quality measure, E-model, and a mean opinion score through the probes. Companies and service providers can use these tools to detect impairments in the network that lower service quality and take immediate action to improve it, such that customers barely notice the change.

The experimental findings demonstrate that the proposed network analysis method surpasses its performance. Moreover, experimental results show the method can effectively perform well in all trace files (new old, normal). In summary, the proposed network analysis method can analyze the network accurately. Table 1 shows a comparison of the proposed method with previous tools (Jtrana, JDNA, NsGTFA). The + symbol indicates that the attribute is working, and the - indicates that it is not working.

Table 1: Proposed Method vs. Previous Tools

Tools/ Methods	**Years**	**Type of trace file**			**Metrics**					**QoS**
		Old	**New**	**Normal**	**Jitter**	**Delay**	**D2D**	**Throughput**	**Goodput**	
NsGTFA	2015	+	+	-	-	+	+	+	-	-
JDNA	2012	+	+	+	+	+	+	+	+	-
Jtrana	2009	+	-	-	+	+	-	+	-	-
New Method	2022	+	+	+	+	+	+	+	+	+

In this Table1 , we note the last two columns. The only method among the comparative methods that extract the quality of services with high accuracy is the proposed new method.

3. QUALITY OF SERVICE THE NETWORK

This Five well-defined network parameters affect the end-to-end quality of transmitted data and, hence, the quality of service. If the path a packet takes from host A to host B over a network can be compared as a set of roads connecting two cities, then QoS attempts to control the Five components of that journey[17],[18].

1. Delay several different kinds of delays may occur in the telecommunications industry, including processing delay, propagation delay, queue delay, and transmission delay. For this, the term "delay" is used interchangeably with "delays," hence the term "end-to-end delay," as in Equation (1)[19], [20].

$$D_{E2E} = D_{processing} + D_{transmission} + D_{propagation} + D_{queuing} \quad (1)$$

Delay strategies may be used for either stalling or counterattacking. Two-way delay (round or time, RTT) is the amount of time it takes for such a packet to traverse in both directions[21], [22].

2. Jitter refers to the variances in one-way delays that individual packets suffer (or variation). Due to its numerous meanings, the phrase has gone out of favor and is now regarded as outdated. The instantaneous packet delay variation may be computed using the one-way delay of two successive packets. as seen in Equation (2):

$$PDV_{\text{Instantaneous}} = D_{n+1} - D_{n} \quad (2)$$

Where D_{n+1} and D_n are one-way delays of two consecutive packets.

Congestion on the network, route alterations, and time drift are all possible reasons for delays. Real-time applications exhibit this well. By using buffering, the effects of delay variation may be mitigated. VoIP connections momentarily store data in a buffer before transmitting it to the other end. As a result, the receiver may arrange the arrival order of incoming packets to preserve as much of the original voice actor stream as feasible. Packet inter-arrival time variation refers to a situation in which data packets sent from the same host arrive at various times (also called a jitter). The equation shows how comparing the arrival times of two successive packets may provide the simultaneous packet inter-arrival time(3):

$$IAT_{\text{Instantaneous}} = A_{n+1} - A_{n} \quad (3)$$

Where A_{n+1} and An are the arrival times of two consecutive packets.

3. The Packet loss, loss of time, and loss of distance are all words used to explain when data intended for host B on network A and B never reaches B. To prevent packets from being held in transit indefinitely, most networks provide a timeout mechanism that causes them to be discarded if they have not arrived at their destination after a certain amount of time. A packet can be designated as lost even if it makes it to its intended destination of B.[23],[24]. Two crucial ideas are inextricably linked to packet loss: the loss of time and the loss of distance. Typically, the longer a sequence of successive packet losses lasts, the more data is lost. From the time a packet is lost to the time the next packet is received, and vice versa is how long the delay is measured. The gap in sequence numbers between two dropped packets stands in for their loss distance. Possible packets have been received at this time. Enter the relevant information into the PLR Calculation Equation (4):

$$PLR=1-(\text{ packets}_{RCV})/(\text{packets}_{SND}) \quad (4)$$

Where packets are located RCV represents the number of successfully received packets, whereas SND represents the total number of packets transmitted.

4. Throughput Refers to the quantity of data that a system can process in a given period. The transmitted and received timestamps of each monitoring packet are stored and monitored by the edge nodes. As a consequence, it is feasible to determine the throughput and the overall traffic distribution among the receiving edge nodes, as in Equation (5) [25].

$$\text{Throughput}= L_i/T \quad (5)$$

Where Li is the packet length and T is the time in seconds.

5. Goodput: The number of bits per second at which a system or network can transfer user data, also known as the users' perceived throughput or application-level throughput (often seconds). By removing the sum of header costs and retransmissions from the throughput, we arrive at the goodput, as in Equation (6) [26], [27].

$$\text{Goodput} = S /T \quad (6)$$

where S is the Size of N in-order packets, and T is the Total receiving time of N packets.

4. PROPOSED METHOD

This analyzer was developed using Java as its programming language. It operates on both Windows and Linux and is platform-independent. And since it's an open-source project, the system's code base may be freely altered by researchers to improve the system's reusability. The system is comprehensive because it does more than just analyze data; it also instructs researchers in the art of calculating network performance metrics, supports all trace file types, and can extract and display graphs without the need for additional tools or scripts. The system's intuitive design makes it easy for anybody to pick up and run with (support usability quality characteristics). In this method, the Java-based NS-2 files analysis proposed algorithm is presented. This algorithm analyzes many files generated from NS-2, these files are represented by computer network scenarios, for example (CN1, CN2..., or CNn). The proposed method can handle almost any trace format produced by NS-2, as well as plotting several graphs for a variety of scenarios and fixing a previous shortcoming of the tool by giving comprehensive data on network performance.

The goal of the proposed is to monitor the representing network by tracking its events. The events (useful information), Throughput, Goodput, Jitter, and End-to-end delay. Therefore, through this tracking of events, the level of performance is evaluated, then, depending on the level of performance, the level of quality of services is determined such as (Poor, Normal and Excellent). However, the useful information is stored in a trace file that is generated from NS-2. The useful information is stored as values with no meaning, pre-processing is required by extracting the important events required in the post-processing. The important events used for monitoring the network performance are by calculating the required measures.

The proposed method is represented three main components. These components elucidate the schema of the main steps of the proposed method. Its comprise processes that are pre-processing steep and post-processing steps. The first, the foundational step is considered where the available information is extricated from the Ns-2 trace file. The post-processing step represents the calculated metrics as static results and can visualize in a two-dimensional manner.

The evaluation step is used to evaluate the performance of the computer networks. The performance will be measured regarding several metrics such as throughput, goodput, Jitter, and end-to-end delay. This step treated is that measured by the similarity between more than one trace file simultaneously. This is strongly required to indicate whether an improvement is achieved or not because it is considered a hard step to monitor multi-trace files in a time series manner. In this case, observations are not mutually independent, file values then a single chance event may affect all later data points. This makes time-series analysis quite different from most other areas of statistics. Thus, this proposed method is a statistical analysis test used to differentiate between multi-trace files. This section consists of three phases (system design, system analysis, and system evaluation).

4.1 System design

The research must initially choose one or even more trace files; when such a "Read Data" option is hit, the NS-2 file is called and analyzed, as seen in Figure 1. The researcher then sets the node ID, packet size, period, and level parameters, as seen in Figures 2 to 5. The researcher then chooses a performance parameter, such as stability or latency, and clicks the "Calculation method" to retrieve the result, as seen in Figures 6 to 10. In these Figures, the network performance results are shown by entering parameters such as period (0.5), packet size (32), ID (0), and level RTR. In light of this, the output process is four network nodes (chosen in the system test), with the total number of packets for the four nodes (16751252), also the number of dropped packets (2067346), and the number of forwarded packets (4534620), and the number of packets that were sent safely (9048268).

Table 1 shows the network parameters, which include four essential components (each node ID, packet size, period, and level). Where specified by (0 to 7). The range is between (32 and 512) while the welt size is between and (32 to 512). (0.5 to 3.0). RTR is the last layer, followed by AGT, CBR, and TCP (ACK). CBR is mostly applied in networking streaming applications because data may be sent across a limited channel capacity. AGT is used to signal the transport layer (e.g., TCP) packet, whereas RTR indicates that the packet has been routed.

Table 2: Simulation parameters

Type of Performance	Node Id	Packet Size	Period	Level
Throughput	0	32	0.5	RTR
Jitter	1	64	1.0	AGT
Goodput	3	128	1.5	TCP
Delay	4	256	2.0	ACK
D-2-D	5	512	2.5	CPR
	6		3.0	
	7			

4.2 System analysis

Due to the network's efficiency, a separate variable may be used to foresee the system's actions. Any simulation is seen as a different manifestation of the system, and its purpose is to investigate and grasp its behavior under a variety of hypothetical conditions. Because this domain is more limited in scope than the actual system, it may provide clues as to what kind of granular restrictions should be placed on the latter. Researchers may test the efficacy and efficiency of the network before executing the modifications in the windows environment. This user interface will make it possible to display all the statistical network information of the trace file that was selected in the previous step. Throughput, as shown in Figures 6-10, is the amount of data transmitted from one endpoint to another during a given time. Throughput is the pace at which data can be sent and received across a network or other means of communication. A bit per second (or bps) is the common measurement of data transfer rates via networks. Congestion occurs in a network when the capacity of the devices connected to it is inadequate to handle the incoming data flow. When this happens, packets are dropped as the devices' buffers fill up. Every packet should get there if there are no problems with the network at the destination. Nonetheless, as the endpoint buffer fills, packets come progressively later. In most cases, the speed of your Internet connection will be faster than your goodput, which is worse than throughput. Latency describes how long it takes for information to arrive at its destination after being sent through a network. The amount of time it takes for data to travel from its origin to its destination and back again is known as the network's latency and is often expressed as a round-trip delay in milliseconds (ms).

4.3 System evaluation

This section evaluates the obtained results based on the speed factor. The results in this section have been checked and compared between the new analyzer and previous work for the Trace graph[28]. We test a new trace file with a total size of 303 Mb. The new analyzer spends ≈ 0.7 minutes to analyze the trace while takes ≈33 minutes in the Trace graph analyzer to complete the analysis for this task as shown in Figure 11.

To prove whether the proposed method shows the results correctly or not, the network analysis tool in Awk language was used for comparison.

Figure 12 and Figure 13, Show the results of the comparison between the proposed method and other network analysis tools using awk language using Throughput event. The results have proven that the proposed method performs the network analysis correctly where the results it is the same in both methods.

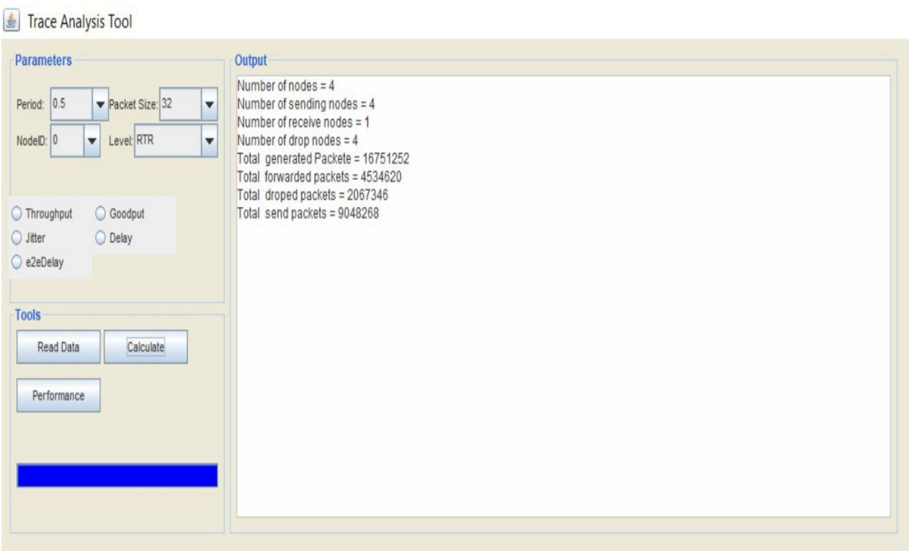

Figure 1. Read Data Trace File From NS-2

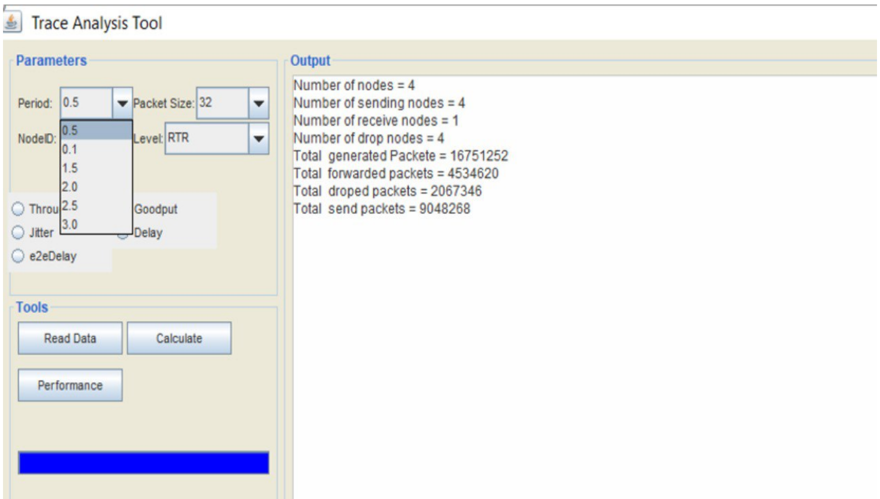

Figure 2. Parameter of Period

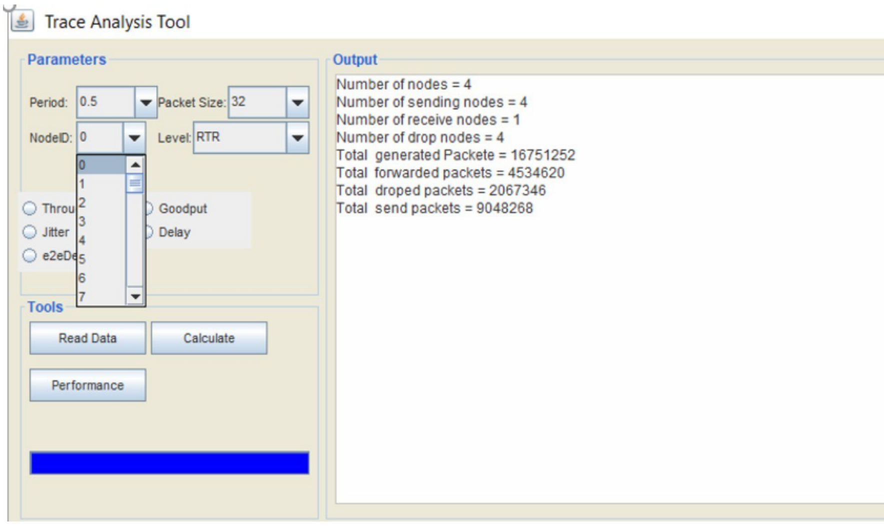

Figure 3. Parameter of the Node ID

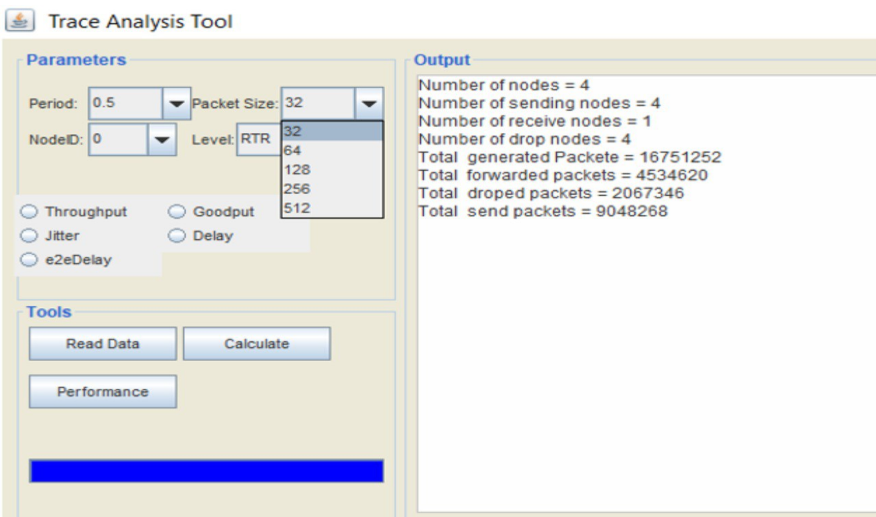

Figure 4. Parameter of Packet Size

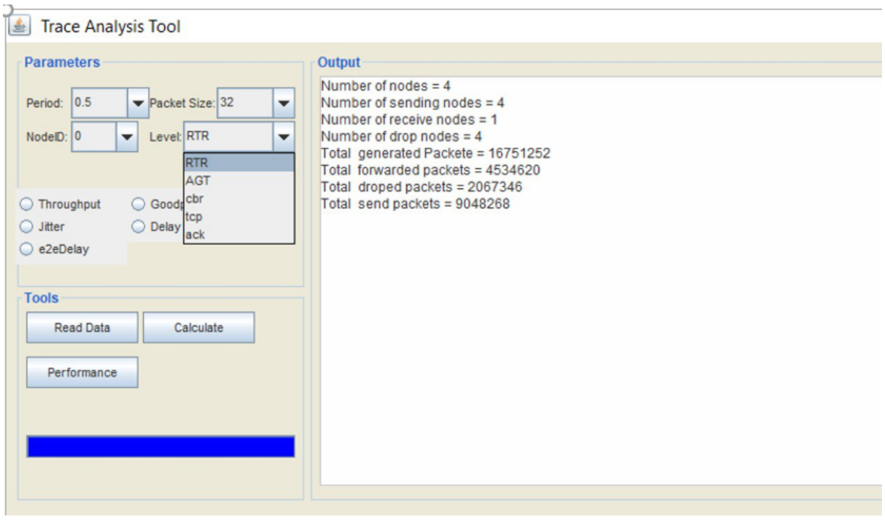

Figure 5. Parameter of Level

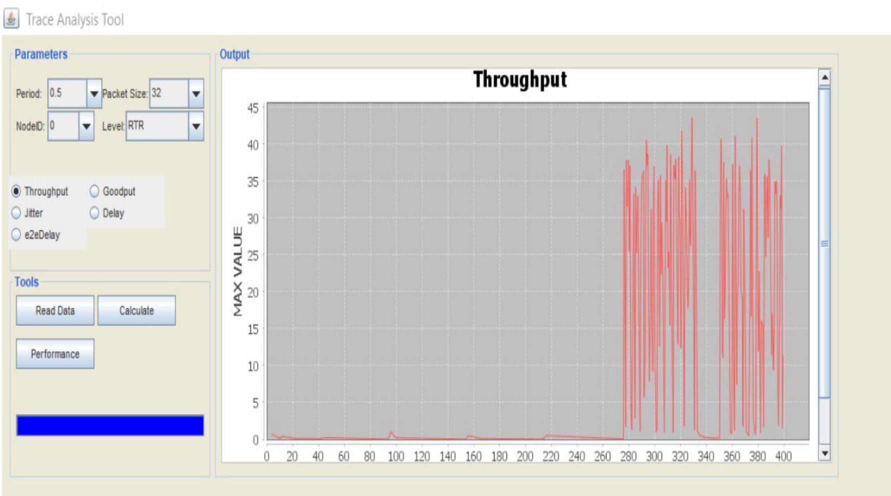

Figure 6. Throughput

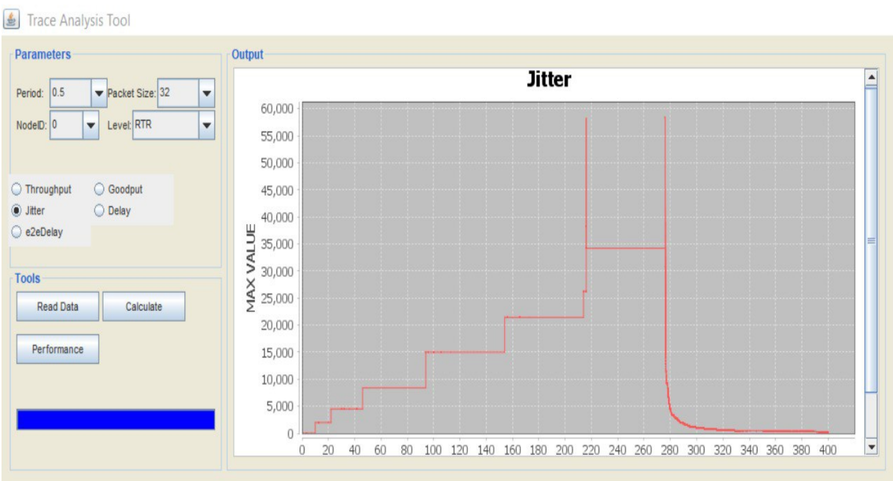

Figure 7. Jitter

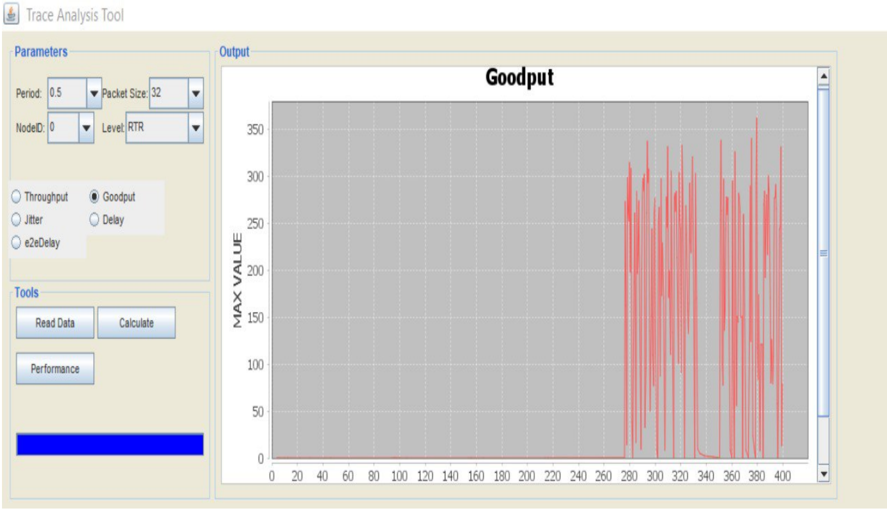

Figure 8. Goodput

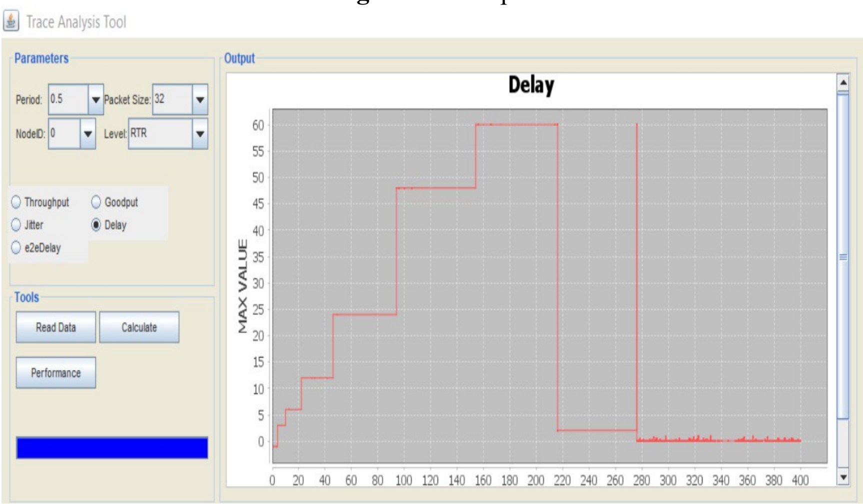

Figure 9. Delay

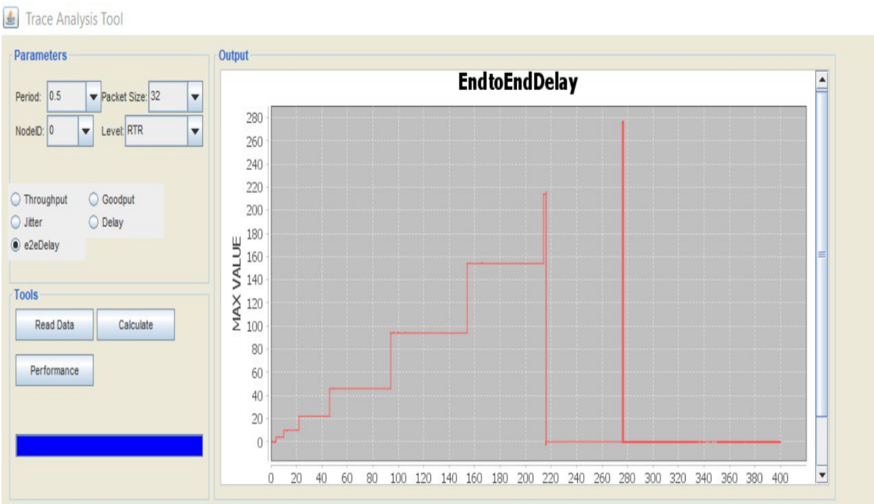

Figure 10. Delay-2-Delay

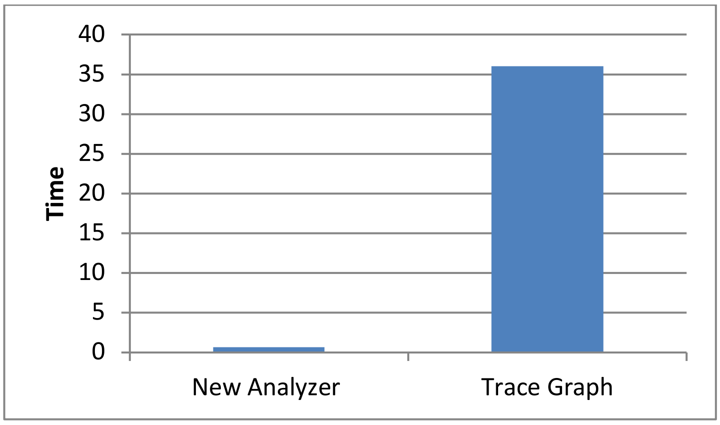

Figure 11. File analysis time

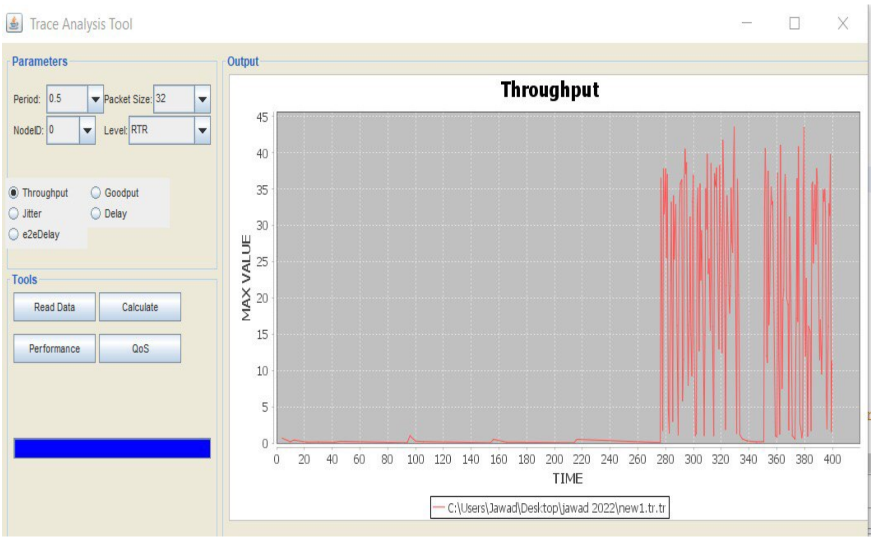

Figure 12. Proposed Method for Throughput Event

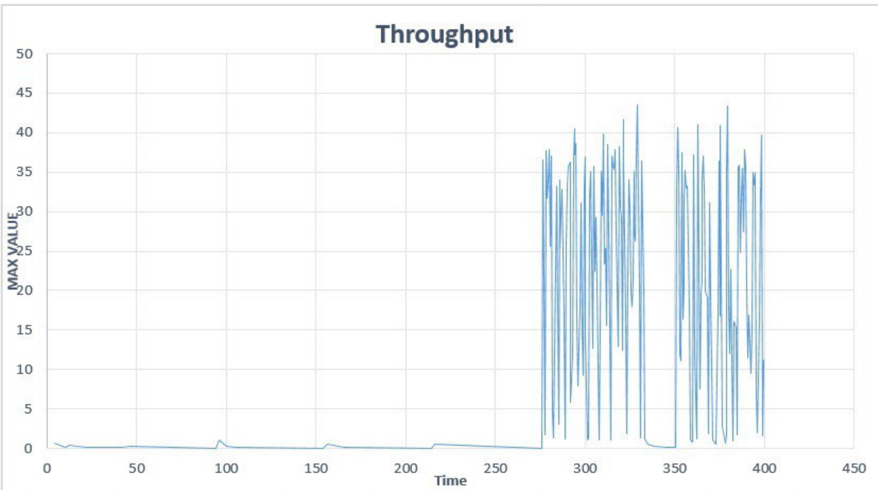

Figure 13. AWK language for the Throughput event

5. CONCLUSION

Network performance is one of the most important problems to resolve, thus the monitoring system must be accurate, user-friendly, and fast enough to disclose network performance in real time. Therefore, all systems, notably in businesses, use Network Simulator NS-2 for network exploration. Following the creation of scripts and scenarios for network simulation, this simulator creates trace files that detail occurrences in the simulated network. Information useful for performance analysis may be found in trace files, such as the total number of packets sent, their arrival times, and any delays or losses along the way. The primary issue is that NS-2 lacks visualization capabilities for analyzing simulation results (tracing files). A new analysis system for tracking files has been proposed, which has proven its worth compared to the Trace graph method. Thus, NS-2 trace analyzers are intended to retrieve performance data. This works with a high-quality Java service, to make it easy to get performance measurements of the output trace file from NS-2 and see network performance. A potential direction for future research in our system is discovered network monitoring is dependent on multi-criteria also known as (multi-objective). Such a notion could draw inspiration from other evolutionary algorithms in real applications, namely genetic algorithms, and genetic programming, which may enhance the performance of the proposed system.

REFERENCES

[1] 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.

[2] J. Alkenani and K. A. Nassar, “Network Performance Analysis Using Packets Probe For Passive Monitoring,” *Informatica*, vol. 46, no. 7, 2022.

[3] J. Alkenani and K. A. Nassari, “Enhance work for java based network analyzer tool used to analyze network simulator files,” vol. 29, no. 2, pp. 954–962, 2023, doi: 10.11591/ijeecs.v29.i2.pp954-962.

[4] C. Nawej, P. Owolawi, and T. Walingo, “Design and Simulation of VANETs Testbed Using OpenStreetMap, SUMO, and NS-2,” in *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, pp. 582–587, 2021.

[5] L.-C. Ku, H.-L. Lee, and M. Chen, “A. Cross-Network Quality and Performance Measurement,” in *2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 1–2, 2021.

[6] H. AL-Behadili, “NTT: Network Topology Tool for Enhancing NS-2,” *Iraqi J. Electr. Electron. Eng.*, vol. 11, no. 1, pp. 101–104, 2015, doi: 10.37917/ijeee.11.1.10.

[7] K. M. Kim *et al.*, “Performance evaluation of maritime VDES networks with OPNET simulator,” in *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, pp. 1–6, 2018.

[8] R. A. Osman, X. H. Peng, M. A. Omar, and Q. Gao, “Quality of service optimisation of device-to-device communications underlaying cellular networks,” *IET Commun.*, vol. 15, no. 2, pp. 179–190, 2021, doi: 10.1049/cmu2.12040.

[9] S. H. Shetty, “Trace File Analysis To Obtain Congestion Window, Throughput And PDR,” *Eur. J. Mol. Clin. Med.*, vol. 7, no. 08, p. 2020.

[10] A. Begum, Z. R. Zahid, and N. Nancy, “Comparative performance evaluation of mobile ad-hoc network routing protocols using NS2 simulator,” *Int. J. Recent Technol. Eng.*, vol. 9, no. 3, pp. 707–713, 2020.

[11] S. Haryadi, “Performance Measurement of Internet Service,” no. March, pp. 1–19, 2018, doi: 10.17605/OSF.IO/8EXMU.

[12] K. K. Ofosu, J.-D. Abdulai, and F. A. Katsriku, “Performance evaluation of Mobile IP on mobile ad hoc networks using NS2,” *Comput. Sci. Inf. Technol. (CS IT)*, pp. 15–27, 2018.

[13] I. S. Ibrahim, P. J. B. King, and H.-W. Loidl, “NsGTFA: A GUI Tool to Easily Measure Network Performance through the Ns2 Trace File,” *J. Intell. Syst.*, vol. 24, no. 4, pp. 467–477, 2015.

[14] S. Chettibi, Y. Labeni, and A. Boulkour, “Trace file analyzer for ad hoc routing protocols simulation with NS2,” in *2015 First International Conference on New Technologies of Information and Communication (NTIC)* , pp. 1–6, 2015.

[15] A. M. M. Habbal, S. Hassan, and A. M. Jabbar, “JDNA: JAVABASED NS-2 ANALYZER,” *Wulfenia J*, vol. 19, no. 9, 2012.

[16] A. U. Salleh, Z. Ishak, N. M. Din, and M. Z. Jamaludin, “Trace analyzer for ns-2,” in *2006 4th Student Conference on Research and Development* , pp. 29–32, 2006.

[17] J. Alkenani, K. A. Nassar, I. Technology, and C. Information, “Network Monitoring Measurements for Quality of Service : A Review,” no. May, 2022, doi: 10.37917/ijeee.18.2.5.

[18] I. S. Alshawi, Z. A. Abbood, and A. A. Alhijaj, “Extending lifetime of heterogeneous wireless sensor networks using spider monkey optimization routing protocol,” *Telkomnika*, vol. 20, no. 1, 2022.

[19] K. A. Nassar and A. A. Abdullah, “End-to-End Fuzzy RED to Reduce Packet Loss in Virtual Circuit Network,” *J. Univ. Babylon*, vol. 25, no. 3, 2017.

[20] D. K. Altmemi and I. S. Alshawi, “Enhance Data Similarity Using a Fuzzy Approach,” *J. Posit. Sch. Psychol.*,

pp. 1898–1909, 2022.

[21] S. Anand and M. V Ramesh, “Performance Analysis of Delay Tolerant Network Routing Protocols in a Heterogeneous Vehicular Network,” in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pp. 1–6, 2018.

[22] T. J. Wong and N. Das, “Modelling and analysis of IEC 61850 for end-to-end delay characteristics with various packet sizes in modern power substation systems,” in *5th Brunei International Conference on Engineering and Technology (BICET 2014)*, pp. 1–6, 2014.

[23] V. Mohan, Y. R. J. Reddy, and K. Kalpana, “Active and Passive Network Measurements : A Survey,” *Comput. Sci. Inf. Technol.*, vol. 2, no. 4, pp. 1372–1385, 2011.

[24] C. W. Ying and L. J. Yu, “Output Feedback Control of Networked Control Systems with Compensate for Packet Loss and Random Time Delays,” in *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 967–971, 2020.

[25] B. Constantine, G. Forget, R. Geib, and R. Schrage, “Framework for TCP throughput testing,” *Internet Eng. Task Force*, vol. 2011, pp. 1–27, 2011.

[26] B. Zhang *et al.*, “Goodput-aware traffic splitting scheme with non-ideal backhaul for 5G-LTE multi-connectivity,” in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2019.

[27] Zhou, Dizhi, Wei Song, and Minghui Shi. "Goodput improvement for multipath TCP by congestion window adaptation in multi-radio devices." *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2013..

[28] K. Lukoit, N. Wilde, S. Stowell, and T. Hennessey, “TraceGraph: Immediate Visual Location of Software Features.,” in *icsm*, pp. 33–39, 2000.