

Enhance work for java based network analyzer tool used to analyze network simulator files

Jawad Alkenani¹, Khulood Ahmed Nassari²

¹Department of Computer Science, Shatt Al-Arab University College, Basra, Iraq

²College of Computer Science and Information Technology, Department of Computer Information Systems, University of Basrah, Basrah, Iraq

Article Info

Article history:

Received Jul 22, 2022

Revised Sep 29, 2022

Accepted Oct 18, 2022

Keywords:

Association

Java language

JDNA

Network performance

Network simulator

Trace files

ABSTRACT

The network performance measurement is important in computer networks, and performance measurement may not be effective for installation in peripheral devices resulting in the replacement of those devices and thus increasing cost. In light of this, it is better to have a simulation of the network to see its performance rather than the actual design. NS-2 is one of the most popular and widely used open-source network simulators in many organizations, which generates trace files during the simulation experience. The trace file contains all network events that can be used to calculate performance. Thus, NS-2 does not offer any visualization options for analyzing simulation results (trace files), which is the fundamental problem of trace file parsing difficulty. This paper provides a graphical user interface tool that enables researchers to quickly and efficiently analyze and visualize NS-2 trace files. This tool is a development of the JDNA tool, as it could not analyze more than one trace file at a time. In addition, this work can be a useful guide for network researchers or other programmers to analyze their networks and understand how to calculate network performance metrics.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Jawad Alkenani

Department of Computer Science, Shatt Al-Arab University College

Basra, Iraq

Email: Jawadalkenani@sa-uc.edu.iq

1. INTRODUCTION

In computer networks, Simulation is one of the most essential methods for studying performance. Simulation enables users to simulate natural systems and provides an overview of the natural system's characteristics and specifications [1]. It enables the use of a different variable to forecast the system's behavior. In all instances, the simulation is viewed as an alternate realization that approximates the system, and its objective is to evaluate and comprehend the system's behavior under many alternative actions or decisions [2]. This field is narrower than the actual system and can identify more precise requirements that could be implemented in the actual system. Before implementing these features to the actual system, the researchers may, for instance, focus on the performance and reliability of the network and present the resulting findings. In addition, networking technologies lower the time and expense required to utilize the natural system [3].

Nowadays, network simulators are utilized by researchers in various domains, including academic education and engineering. By assessing their system through network simulation, engineers can create and model a new system to obtain performance. In addition, it can be used to evaluate the effect of the various parameters and investigate the system's unique behavior [4]. In general, network simulation encompasses a vast array of network technologies and protocols. It facilitates the construction of complex networks from fundamental building pieces such as sets of nodes and links. Users can construct unique network topologies

utilizing various node types, including end-hosts, hubs, network bridges, routers, and optical link-layer devices, to create diverse network topologies. In this light, the NS-2 is primarily used for network research [5]. It is widely around the world. However, NS-2 does not offer any visualization choices for simulation results (trace files) analysis; the trace file contains all network events that can be utilized to derive network performance measures. However, the data supplied by NS-2 is difficult to interpret independently, and NS-2 does not provide sufficient graphical results for a simulation trace file [6]. Consequently, the network researcher takes more time reading the trace file and comprehending other script tools, which makes the simulation procedure more difficult [7]. It would be more efficient if there was a newly created tool that could quickly extract and present the result via an interactive graphical user interface (GUI). Using this tool will allow the user to get rid of JDNA limitations (the user cannot select more than one trace file at the same time). In this work, we study the problem of designing a low-cost passive monitoring system to compute a network performance metric. The proposal is a Java-based system that analyzes NS2 files and extracts important network events, after the procedures for detecting and processing the trace file, the system presents the information contained in the trace files via a GUI along with a set of simulation-related data, metrics, and statistics.

This paper is organized as follows: In section 2, previous work is presented on how to analyze network performance. Section 3 presents the metrics used to measure network performance. Section 4 describes a suggested tool for analyzing NS2 trace files using Java. In section 5, the tool is analyzed. Finally, the conclusion of this paper is presented in section 6.

2. RELATED WORK

Several researchers have highlighted quality monitoring and network performance evaluation. The authors proposed in [8], is to improve healthcare for patients to minimize delay and packet loss. The network measurement was calculated based on the congestion window of transmission control protocol (TCP). The analysis of all parameters is calculated based on the output of the simulator which is the trace file. However, the proposed study doesn't show the network format supported and the statistical results are calculated in external tools. Begum *et al.* [9], evaluate the performance of several mobile ad-hoc network routing protocols based on several metric measurements. Based on four crucial measures, including packet delivery ratio, average end-to-end delay, normalized routing overhead, and throughput, the evolution's performance varies with the number of mobile nodes and packet sizes. However, the authors use script language to perform all the experiments which makes the process of the evaluation require more time with understating other script languages. Ofosu *et al.* [10], evaluate mobile internet protocol (IP) on mobile ad hoc networks based on the evolution metrics. Similar to the above research, the authors used an external scrip language to perform the analysis which in the end extra time was required. It will be more effective if there is a tool that can be used to evaluate the performance with an interesting graphical interface that makes the work easy and understandable. Ibrahim *et al.* [11], there is a lack in the way of network measurement and limitations in the statistics and graphics reports provided. Although, the authors presented a new tool that was able to analyze the trace file with many statistical reports. However, many parts are limited and required attention such as the proposed tool doesn't run in all operating systems, and being unable to read multi-files. Chettibi *et al.* [12], this research was able to analyze the output of the simulator with several graphical interfaces. However, it is limited in processing all types of trace files and unable to read multiple trace files simultaneously. Habbal *et al.* [13], presented an interface tool able to present several reports and statically results.

The researchers can monitor the network performance to decide whether it is good or not. However, although the tool can present several metrics it is limited only to reading two trace files only. Lehikoinen and Rätty [14] challenging in the field of video streaming because of its sensitivity to transmission quality variations. In streaming applications, the ability to quantify the experienced quality of service (QoS) is valuable. By employing the information from QoS measurements, video traffic can be altered to fit the transmission limits of the network, hence enhancing the perceived playback quality. The purpose of this project is to review the notion of QoS, investigate various QoS monitoring methodologies, and develop a system that monitors the end-to-end QoS of several simultaneous video streaming sessions. The research is based on the constructive technique of relevant publications and technologies, and the results are generated from the end-to-end QoS monitoring system that has been created. Salleh *et al.* [15], presented an interface tool able to analyze network performance showing the quality of the networks. However, this work doesn't support all kinds of network formats. In addition, the tool can process one trace file only which analyzes different configurations harder. Agrawal *et al.* [16], They present both active and passive probes that allow service providers to detect service disruptions. We use the probes to compute the network parameters (delay, loss, and jitter) that may be used to compute the call quality using a voice quality metric, E-model, and a Mean Opinion Score. Service providers and corporations can use these technologies to discover network impairments that degrade service quality and take corrective actions in real time so that end-users perceive the degradation as minimal.

3. NETWORK PERFORMANCE ANALYSIS

Network performance analysis, as the name implies, is the use of network data to decipher performance trends. In this work, we use the most popular metrics related to performance analysis [17], [18].

- a) Delay (latency): processing delay, propagation delay, queue delay, and transmission delay are all examples of delay that can occur in the telecommunications industry. In this work, the term "delay" refers to any and all delays, hence the term "end-to-end delay" was used to describe it, as in (1), [19].

$$DE2E = D_{processing} + D_{transmission} + D_{propagation} + D_{queuing} \quad (1)$$

The intent of any delay tactics could be either to stall or to counteract. While "one-way delay" refers to the amount of time it takes for a packet to go from the sending host to the receiving host and back again, "two-way delay" (or round-trip time, RTT) describes the amount of time it takes for the packet to travel both ways [20].

- b) Jitter: it refers to the differences in the one-way delays experienced by individual packets (or variation). Due to its multiple interpretations, the term has fallen out of use and is now considered obsolete. One-way delays of two consecutive packets can be used to calculate the instantaneous packet delay variation. packets as in (2) [21].

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

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

Congestion on the network shifts in the path taken, or timing drift are all potential causes of delays. Real-time programs clearly display this. The consequences of delay variation can be mitigated with the help of buffering. Voice over internet protocol (VoIP) connections use a buffer to temporarily store data before playing it back to the other end, which helps the receiver arrange and space incoming packets so that as much of the original voice stream is preserved as possible [22]. A host's packets may arrive at different times, a phenomenon known as packet inter-arrival time variation (also called a jitter). The instantaneous packet inter-arrival time is calculated by comparing the arrival times of two consecutive packets, as shown in (3):

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

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

- c) The terms "packet loss," "loss time," and "loss distance" are all used to describe the situation in which a packet is sent from host A to host B but never reaches B. Since it would be impracticable to wait forever for a packet, networks typically include a timeout mechanism that causes the packet to be dropped if it takes too long to reach the other end of the network. A packet may be marked as lost even if it eventually reaches destination B [23], [24].

There are two crucial ideas that are intrinsically tied to packet loss: loss time and loss distance. When a packet loss event occurs, the amount of loss time is equal to the total number of consecutive packets lost. Time is measured from the moment one packet is lost till the moment the next packet is received, and vice versa. The sequence number difference between two lost packets represents the loss distance between them. It's possible that they got some packets in the interim. To determine the PLR, plug the values into the (4):

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

where $\text{packets } RCV$ denotes the amount of correctly received packets and $\text{packets } SND$ the total sent packets.

- d) Throughput :Means the amount of information that can be processed by a system in a given length of time. Each monitoring packet's sent and received timestamps are recorded and tracked by the edge nodes. As a result, it is possible to calculate the throughput and the total traffic distribution between the edge nodes that are receiving the traffic, as in (5) [25].

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

where L_i is the length of packet i , and T is the time in seconds.

- e) Goodput: Users' perceived throughput, also known as application-level throughput, is what this term refers to, goodput is the number of bits per second at which a system or network can transmit user data (often seconds). Goodput can be calculated by taking throughput and then subtracting the total amount of header expenses and retransmissions, as in (6) [26].

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

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

4. THE PROPOSED TOOL

The proposed tool tries to develop a JDNA analyzer [13], this analyzer method can process most trace format that is generated by NS-2, the plot more than graphs to a different scenario, And also the treatment of the problem he was suffering from, which was not analyzing more than file as well as, providing all information required for network performance. To design the architecture of an NS2 trace file analyzer using Java, and before starting to write the actual code, it is necessary to understand what will be delivered to the user. The specifications for the needs of the previous stage should match what we intend to design. Therefore, this stage is followed by the requirements analysis stage to build typical interfaces. System design helps determine hardware and system requirements, as well as the overall architecture of the system. The system design specification is the input for the later modelling phase [27]. The proposed tool divides into two steps as shown in Figure 1.

The network performance measurement is according to a novel statistical test. Such a test is in use widely by the practitioners of machine learning validation. In this statistical test (i.e., t-test), the experimental results display that the performance of any network scenario is determined based on the metrics for monitoring service quality (e.g., Jitter and Delay). This statistical test is a type of inferential test that determined if there is a significant difference based on the factors of two networks, which may be related to each another on certain features. The statistical test can guarantee a confident result, thus the result is real, not just got based on luck for each factor being tested.

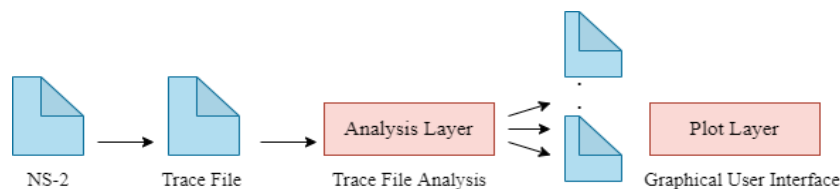


Figure 1. Tool analyzing an NS2 trace file

4.1. Analysing layer

The analyzing layer extracts and filters key fields from the NS-2 trace file to classify the captured data and generate categorizing distinct metrics files, such as "Throughput file", "Goodput file", and "Jitter file". These created files will be stored in a certain location on the client's computer. The file sizes are significantly smaller than the original trace file generated by NS-2. The relevance of this procedure is to tie the reading and calculation of network performance measurements to these files, hence accelerating post-processing.

4.2. Plot layer

The plot layer calculates the result generated by the layer behind it. It generates graphs and generates reports based on user input. The user can enter a specified value such as node Id, source, destination, flow Id, and the period to plot various network performance metrics including "Jitter result", "Throughput result", and "Goodput result". The plot layer includes extra experiment-related information. It provides additional information like the duration, number of available nodes, number of sending nodes, and number of receiving nodes.

5. PERFORMANCE EVALUATION

The tool is built using the programming language Java, it is platform-independent and can operate on both Windows and Linux. Additionally, the method is an open-source tool that permits researchers to freely change and update source code (supports reusability quality characteristic). In addition, it teaches researchers how to calculate network performance measures. In addition, the system supports all trace file types and can easily extract and plot graphs without requiring any additional tools or scripts; it is a comprehensive analyzer. The system features an intuitive design, making it very simple to use (support usability quality characteristic). The proposed network monitoring tool can be used in real networks' performance evaluations, and it enables experts to review the result due to the simplicity of the graphical representation. In this way, the proposed system provides a deep insight into the network factors that lead to determining the glitch. For example, in the network, many metrics affect QoS. Therefore, finding the specific factor is very important and can assist in the enhancement of the QoS.

Figure 2 shows how to choose more than one file to perform the performance analysis process. This interface allowed the user to choose their trace files. The user must select "File" from the menu bar and then "open" from the drop-down menu. The system displays the "choose data file window," in which the user must indicate the location of the trace file on his or her computer. After selecting the file, his system displays in Figure 3 a message containing information about the trace file. This information displays the kind of trace file,

the total number of data packets in the file, and the amount of time spent reading and analyzing the trace file. Table 1 shows the network parameters that contain four basic components (each node ID, packet size, period, and level). Where represented as follows, the node ID is from (0 to 7). The welt size ranges from (32 to 512), and the period runs from (0.5 to 3.0). Finally, the level is routed level (RTR), transport layer (AGT), constant bit rate (CBR), transmission control protocol (TCP), an acknowledgment (ACK). CBR is mainly used in networking streaming applications as content can be transferred through a limited channel capacity, it is AGT to indicate the transport layer (e.g. TCP) packet, or RTR if it concerns the routed packet.

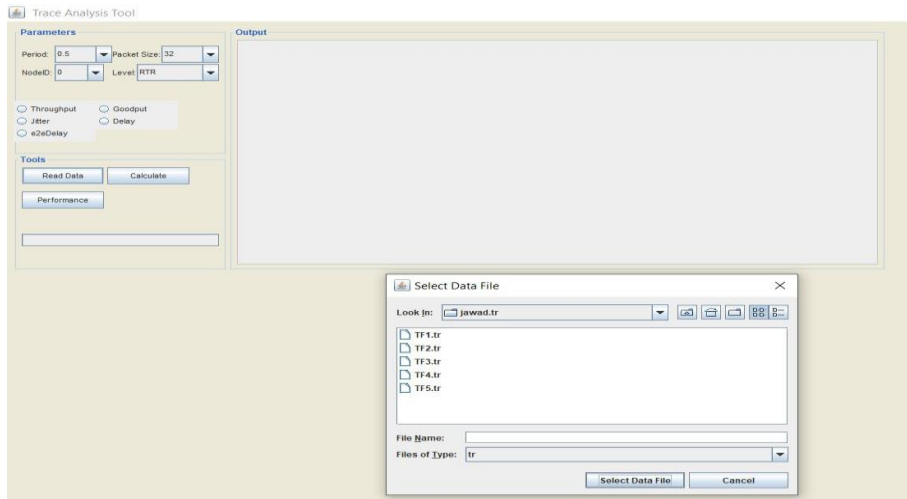


Figure 2. Select the data trace files from NS-2



Figure 3. Read data trace files from NS-2

Table 1. Trace analysis tool

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
Delay	5	512	2.5	CPR
-2	6		3.0	
-Delay	7			

The researcher must first select one or more trace files, the NS-2 file is called and analyzed when the "Read Data" button is clicked. Second, the researcher defines the parameters of the node ID, packet size, period, and level. The researcher then chooses a performance parameter, such as instability or lag, and clicks the "Calculate" button to get the result, as shown in Figures 4-8.

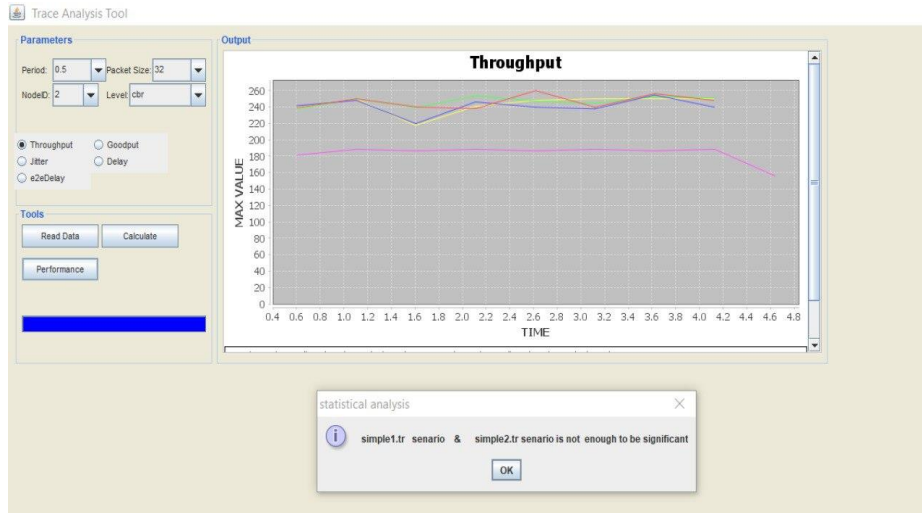


Figure 4. Throughput

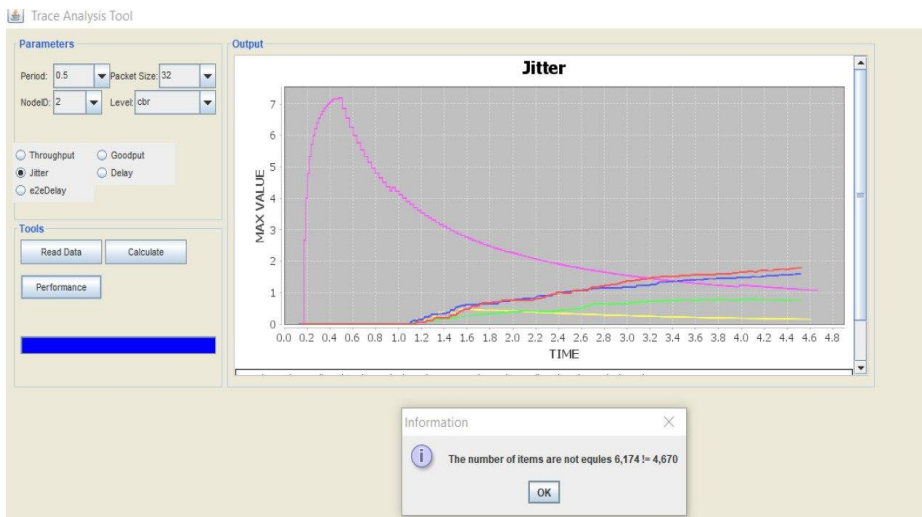


Figure 5. Jitter

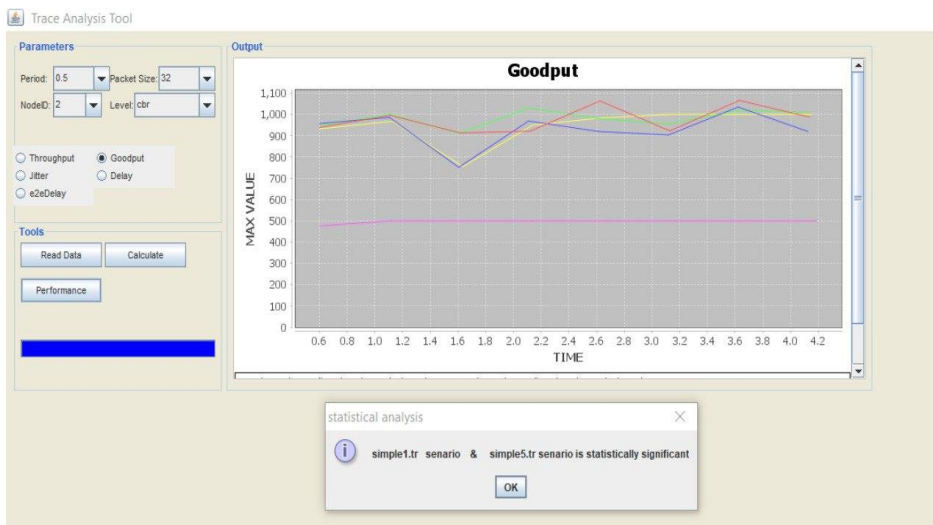


Figure 6. Goodput

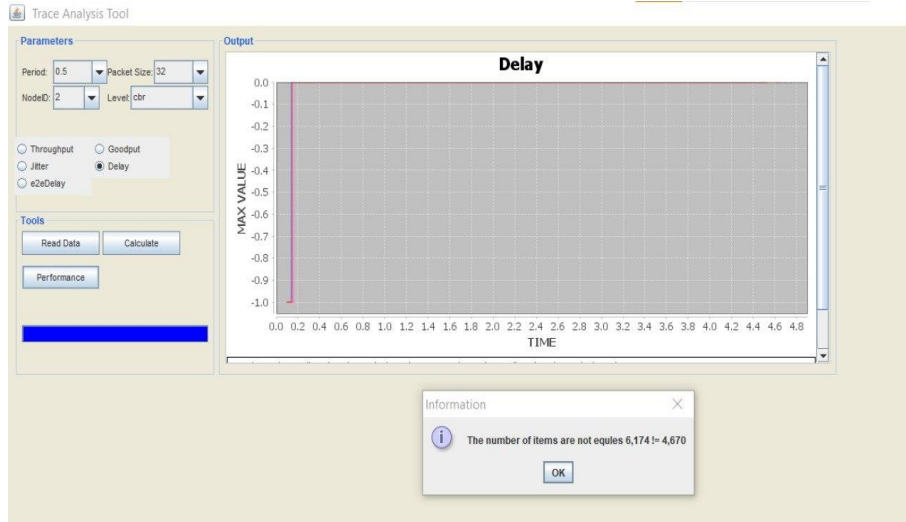


Figure 7. Delay

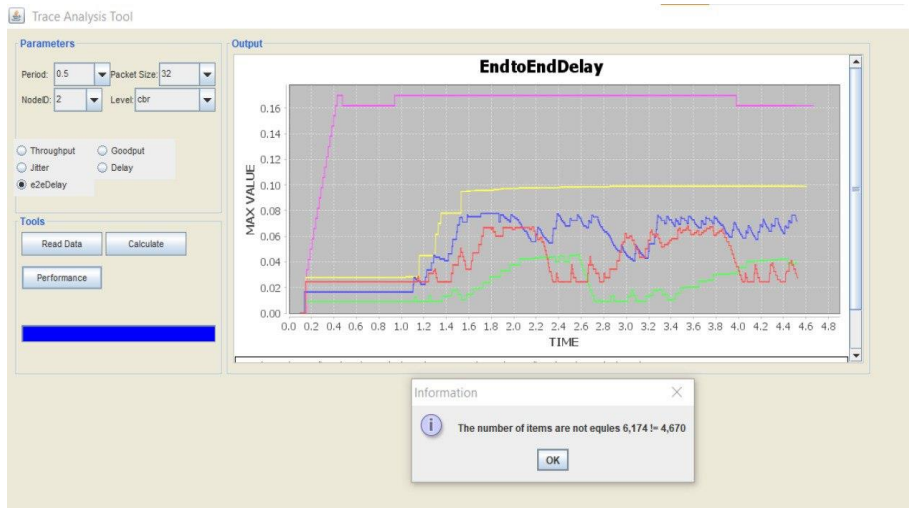


Figure 8. Delay-2-Delay

Network performance is measured against a statistical test in this statistical test (eg, t-test), empirical results show that the performance of any network scenario is determined by QoS monitoring measures (eg, Goodput). This statistical test is a kind of inferential test that determines whether there is a significant difference based on the factors of five networks, which may be related to each other in certain features. Figure 9 shows, a comparison of the five files (belonging to 5 networks), where we observe a different p-value Very between the first and fifth network.

The purpose of this step is to indicate the best or the highest performance produced among multi files. This step is so important because it is hard to analyze the performance of multi-series in different times. Thus, in this step statistical analysis of paired samples T-test is used to monitor multi files and indicates the best quality them. The statistical analysis of paired samples T-test is performed to test if the results of metrics such as example throughput are significantly different according to the mean of each metric according to the file that belongs. The null hypothesis known as H_0 indicates there is no difference between two metrics based on the mean result, while H_1 is the alternative hypothesis representing the difference between two mean metrics results. The rule can be simplified into two states even acceptance or rejection. The acceptance and rejection are based on the p-value obtained in the statistical analysis where less than 0.05 indicates a sufficient difference between the means of the metrics.

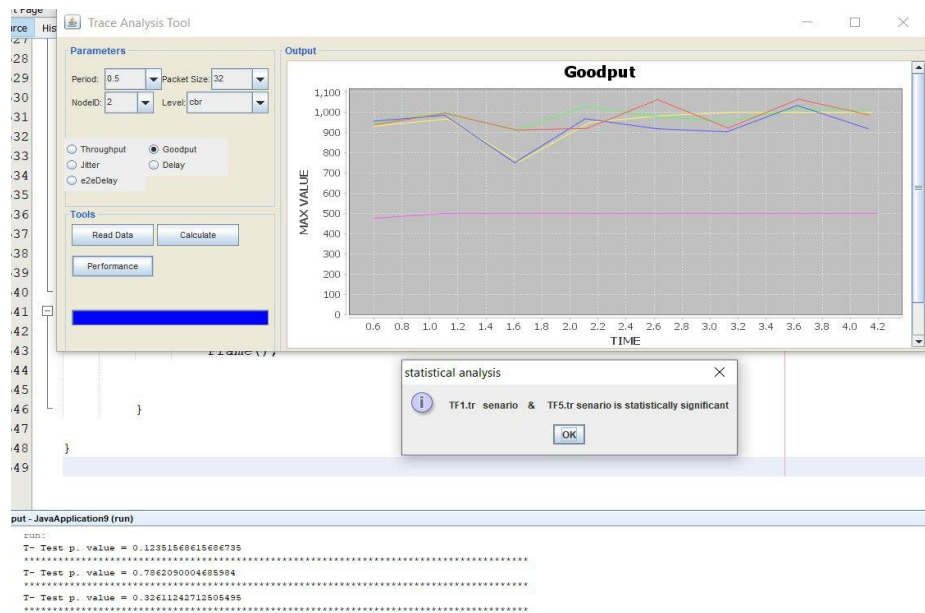


Figure 9. P-Value between of the network

6. CONCLUSION

Operators globally must guarantee everything operates well and monitor the network performance. The monitoring system must be accurate, easy to use, and fast enough to show real-time network performance. Service quality depends on network performance. Network simulator NS-2 is used for network research on UNIX and Windows systems, especially at institutions. After network simulation scripts build network scenarios, trace files record network events. Trace files capture data for performance analysis, such as the number of packets sent from source to destination, packet delay, and packet loss. NS-2 lacks visualization options for trace file analysis. This paper provides a graphical user interface tool that enables researchers to quickly and efficiently analyze and visualize NS-2 trace files. This tool is a development of the JDNA tool, as it could not analyze more than one trace file at a time. In future work, it will be more interesting if we improve this tool so that we can determine the overall network quality in each file.




REFERENCES

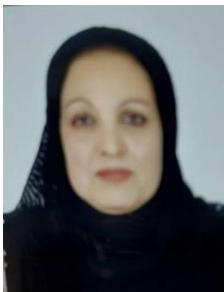
- [1] N. Effendy *et al.*, "The elastic, mechanical and optical properties of bismuth modified borate glass: Experimental and artificial neural network simulation," *Opt. Mater. (Amst.)*, vol. 126, p. 112170, 2022, doi: 10.1016/j.optmat.2022.112170.
- [2] T. C. S. Ponraj, R. Sukumaran, S. R. Vignesh, T. T. Manikandan, and M. Saravanan, "Analysis on simulation tools for underwater wireless sensor networks (UWSNs)," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, pp. 207–211, 2021, doi: 10.1109/ICICV50876.2021.9388401.
- [3] 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, 10.1109/ISPACS51563.2021.9650987.
- [4] 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, doi: 10.1109/CSNDSP.2018.8471848.
- [5] F. Morales, Y. Luzón, E. González, and D. Tipán, "Performance analysis of a mobile ad-hoc network (MANET) under IEEE 802.11 n and IEEE 802.11 ac standards using NS-2 simulation tool," *International Conference on Information Technology & Systems*, Springer, Cham, 2022, doi: 10.1007/978-3-030-96293-7_1.
- [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] B. G. Gamiya, A. Adebisi, S. S. Umar, O. Oshiga, S. Thomas, and F. Amu, "Power quality improvement for residential customers of distribution network through monitoring," In: *2021 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*. IEEE. p. 1-5, 2021, doi: 10.1109/ICMEAS52683.2021.9692401.
- [8] S. H. Shetty, "Trace file analysis to obtain congestion window, throughput and PDR," *Eur. J. Mol. Clin. Med.*, vol. 7, no. 08, p. 2020.
- [9] 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, doi: 10.35940/ijrte.C4691.099320.
- [10] 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, doi:10.5121/csit.2018.80302.
- [11] 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, doi: 10.1515/jisys-2014-0153 10.1515/jisys-2014-0153.




- [12] 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, doi: 10.1109/NTIC.2015.7368739.
- [13] A. M. M. Habbal, S. Hassan, and A. M. Jabbar, "JDNA: Javabased NS-2 Analyzer," *Wulfenia J*, vol. 19, no. 9, 2012.
- [14] L. Lehikoinen and T. Rätty, "Monitoring end-to-end quality of service in a video streaming system," *Proc. 2009 8th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2009*, 2009, pp. 750–754, doi: 10.1109/ICIS.2009.167.
- [15] 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 Developmen*, 2006, pp. 29–32, doi: 10.1109/SCORED.2006.4339302.
- [16] S. Agrawal *et al.*, "VoIP service quality monitoring using active and passive probes," In: *2006 1st International Conference on Communication Systems Software & Middleware*. IEEE, p. 1-10, 2006, doi: 10.1109/COMSWA.2006.1665188.
- [17] J. Alkenani, K. A. Nassar, I. Technology, and C. Information, "Network Monitoring measurements for quality of service : A review," *network*, vol. 2, no. 3, 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 (Telecommunication Computing Electronics and Control)*, vol. 20, no. 1, 2022, doi: 10.12928/telkonnika.v20i1.20984.
- [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] 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 (ICIC)* 2018, pp. 1-6, doi: 10.1109/ICIC.2018.8782333.
- [21] 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)*, 2014, pp. 1–6, doi: 10.1049/cp.2014.1073.
- [22] D. K. Altmemi and I. S. Alshawi, "Enhance data similarity using a fuzzy approach," *J. Posit. Sch. Psychol.*, pp. 1898–1909, 2022.
- [23] G. Cannone, S. Moscato, and M. Oldoni, "Passive networks for c-band multi-carrier wireless backhaul systems," *2021 IEEE MTT-S International Microwave Filter Workshop (IMFW)*. IEEE, p. 275-27, 2021, doi: 10.1109/IMFW49589.2021.9642278.
- [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)*, 2020, pp. 967–971, doi: 10.1109/ICISCE50968.2020.00199.
- [25] M. Harayama and N. Miyagawa, "Intelligent throughput stabilizer for UDP-based rate-control communication system," *Intelligent and Converged Networks*, pp. 205-212, 2021, doi: 10.23919/ICN.2021.0014.
- [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)*, 2019, pp. 1–6, doi: 10.1109/WCNC.2019.8885728.
- [27] T. J. Halloran, "The fog of software design," *IEEE Softw.*, vol. 38, no. 3, pp. 132–135, 2021, doi: 10.1109/MS.2021.3056937.

BIOGRAPHIES OF AUTHORS



Jawad Alkenani    is a master student in in College of Computer Science and Information Technology, Computer Information Systems Department, University of Basrah, Basrah, Iraq. He is Interested in computer networks and data mining. He can be contacted at email: Jawadalkenani@sa-uc.edu.iq.



Khulood Ahmed Nassari    is she doctor in College of Computer Science and Information Technology, Computer Information Systems Department, University of Basrah, Basrah, Iraq. She is Interested in computer networks. She can be contacted at email: khulood.nassar@uobasrah.edu.iq.