A Comparative Study Between Wireshark and Paessler Router Traffic Grapher (PRTG) in Network Monitoring and Analysis.

Anis Nariesha Alizan[1*], Mohamad Fadli Zolkipli[1]

[1]The School of Computing (SOC), College of Arts and Science (CAS),
Universiti Utara Malaysia (UUM), Sintok, 06010 Bukit Kayu Hitam,
Kedah, Malaysia.

***Email:** anis_nariesha_a@soc.uum.edu.my

Abstract

Network traffic analysis (NTA) plays a major role for monitoring network performance, improving security, and guaranteeing operational efficiency across sectors. This study compares two popular network analysis tools, Wireshark and Paessler Router Traffic Grapher (PRTG), that handle distinct areas of network monitoring. Wireshark focusses on packet-level analysis, making it ideal for comprehensive troubleshooting and security protocol analysis. PRTG, on the other hand, provides a comprehensive picture, monitoring a wide range of network devices and allowing for scalability via distributed monitoring, making it ideal for organizations with complicated infrastructure. This study assesses each tool's capability using essential criteria such as scalability, real-time monitoring, data management, and security features. By understanding the strengths and limitations of Wireshark and PRTG, this paper aims to assist network professionals in choosing the most suitable tool for their specific monitoring and diagnostic requirements. Practical recommendations are provided to guide both beginners and experienced users in leveraging these tools effectively.

Keywords

Network Management, Network Monitoring, Network Traffic Analysis (NTA), Packet-level Analysis, Paessler Router Traffic Grapher (PRTG)

Introduction

Network traffic monitoring and analysis (NTA) is vital for detecting security threats and maintaining network health (Kim & Sim, 2019), especially as industries increasingly rely on stable connectivity for operations and data access. By continuously tracking both real-time and historical network activity, NTA identifies performance bottlenecks, spots usage trends, and reveals vulnerabilities such as weak protocols or suspicious behaviour (So-In, 2006). This proactive monitoring enhances security, facilitates debugging, and prevents network failures that could disrupt access to resources. Wireshark and Paessler Router Traffic Grapher (PRTG) are popular tools for these objectives, with Wireshark excelling at precise packet-level analysis and PRTG

Submission: 1 October 2024; **Acceptance:** 14 November 2024

providing full infrastructure monitoring. Comparing these solutions reveals vital insights into their strengths and enables network professionals to select the best tool for their specific needs, eventually enhancing network performance, security, and administration.

The main goal of this paper is to perform a comparison of Wireshark and PRTG by highlighting their capabilities and applications. This journal will point out each tool's crucial role in network monitoring and management by exploring their strengths and limitations in hope to provide a deeper insight on how these tools operate and to aid users in choosing the best tool that meets their needs. Next, this paper aims to provide a practical recommendation tailored for both beginners and professionals in network monitoring and analysis. By exploring each tool, this journal aims to provide beginner-friendly guidance that can be basics for students or entry-level IT professional who are just begun to study network monitoring. The organization of this paper is outlined as follows.

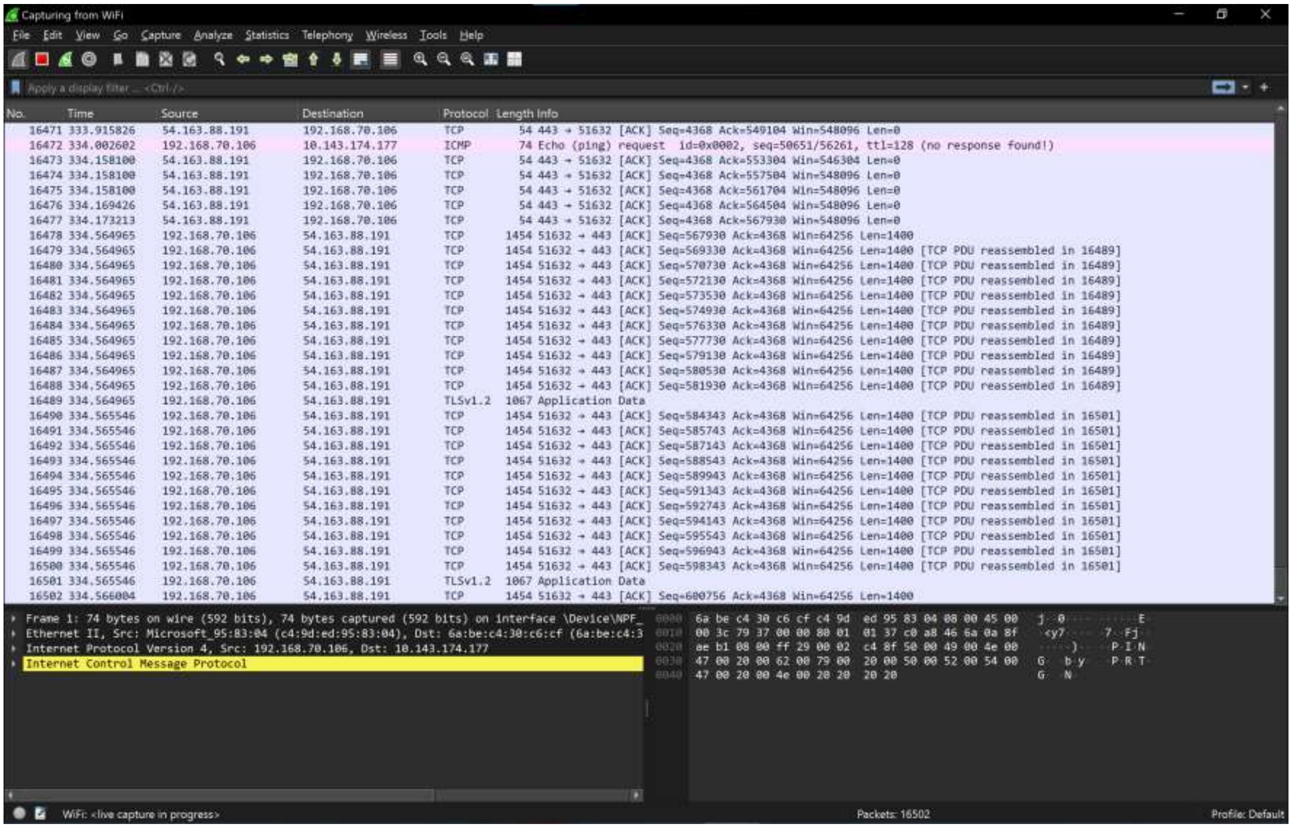
Section 3 presents a literature review that delves into each tool individually, covering their introductions, key features, applications, and strengths and weaknesses to set the stage for the comparison. Section 4 conducts a comparative analysis based on critical criteria such as scalability, real-time monitoring, data handling capacity, complexity, and security features. This section also examines each tool’s performance in small and large networks, resource efficiency, and cost-effectiveness. Section 5 addresses challenges in using Wireshark and PRTG, shedding light on potential difficulties encountered by users. Section 6 offers practical recommendations for selecting the most suitable tool based on network analysis requirements. Finally, Sections 7 and 8 provide acknowledgements and references, respectively, recognizing supporting resources and foundational studies.

- **Wireshark**

Wireshark is a popular open-source tool designed for network analysis, enabling the capture and visualization of real-time network traffic, helping network administrators troubleshoot issues, monitor protocols, and enhance security (Mala et al., 2023). Wireshark's packet capture (PCAP) capability converts raw network data into a readily readable format, allowing users to quickly identify bottlenecks, latency difficulties, and connectivity problems. The tool's real-time analysis and filtering capabilities provide rapid insights, allowing for effective targeting of specific traffic kinds or protocols. Wireshark's graphical user interface (GUI) is simple to use, making it suitable for both new and expert users. Wireshark is widely valued for its thorough packet analysis, protocol support, and utility in activities such as quality of service (QoS) monitoring and network forensics, which use historical data to examine security breaches and compliance.

Despite its numerous benefits, Wireshark has a learning curve, as its advanced features might be intimidating for newbies. It is restricted to recording traffic on networks physically accessible to the device it is operating on, making remote monitoring difficult without additional configuration. Wireshark can also be resource heavy, especially during long sessions or when dealing with huge amounts of data, which might have an influence on system performance. Furthermore, there are security risks, as Wireshark's thorough packet access may reveal critical information if not handled securely. It also has difficulties with encrypted traffic because it cannot decrypt data without the necessary encryption keys, limiting its use in fully encrypted contexts.

Figure 1 shows the interface of Wireshark, which is a capture session monitoring network traffic on a Wi-Fi interface. Each line in the graphic represents a data packet (a small bit of information) sent over the network, including where it came from, where it is going, the type of data, and some technical details. Most packets here use TCP, a popular protocol for ensuring stable communication. Some packets indicate problems, such as the need to be resent or joined with others, which could be due to delays or network failures. One highlighted line depicts a different form of packet (ICMP), which is commonly used for diagnostic purposes and received no response, indicating that the message was not received. The bottom portion breaks down the packet into technical specifics, demonstrating how it appears in its raw, coded form.



Capturing from WiFi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.     Time        Source          Destination     Protocol Length Info
  16471 333.915826  54.163.88.191   192.168.70.106  TCP        54 443 → 51632 [ACK] Seq=4368 Ack=549104 Win=548096 Len=0
  16472 334.002602  192.168.70.106  10.143.174.177  ICMP       74 Echo (ping) request  id=0x0002, seq=50651/56261, ttl=128 (no response found!)
  16473 334.158100  54.163.88.191   192.168.70.106  TCP        54 443 → 51632 [ACK] Seq=4368 Ack=553304 Win=546304 Len=0
  16474 334.158100  54.163.88.191   192.168.70.106  TCP        54 443 → 51632 [ACK] Seq=4368 Ack=557504 Win=548096 Len=0
  16475 334.158100  54.163.88.191   192.168.70.106  TCP        54 443 → 51632 [ACK] Seq=4368 Ack=561704 Win=548096 Len=0
  16476 334.169426  54.163.88.191   192.168.70.106  TCP        54 443 → 51632 [ACK] Seq=4368 Ack=564504 Win=548096 Len=0
  16477 334.173213  54.163.88.191   192.168.70.106  TCP        54 443 → 51632 [ACK] Seq=4368 Ack=567930 Win=548096 Len=0
  16478 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=567930 Ack=4368 Win=64256 Len=1400
  16479 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=569330 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16480 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=570730 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16481 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=572130 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16482 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=573530 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16483 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=574930 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16484 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=576330 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16485 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=577730 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16486 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=579130 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16487 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=580530 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16488 334.564965  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=581930 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16489]
  16489 334.564965  192.168.70.106  54.163.88.191   TLSv1.2  1067 Application Data
  16490 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=584343 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16491 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=585743 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16492 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=587143 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16493 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=588543 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16494 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=589943 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16495 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=591343 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16496 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=592743 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16497 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=594143 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16498 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=595543 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16499 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=596943 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16500 334.565546  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=598343 Ack=4368 Win=64256 Len=1400 [TCP PDU reassembled in 16501]
  16501 334.565546  192.168.70.106  54.163.88.191   TLSv1.2  1067 Application Data
  16502 334.566004  192.168.70.106  54.163.88.191   TCP      1454 51632 → 443 [ACK] Seq=600756 Ack=4368 Win=64256 Len=1400
```

```
› Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_
› Ethernet II, Src: Microsoft_95:83:04 (c4:9d:ed:95:83:04), Dst: 6a:be:c4:30:c6:cf (6a:be:c4:3
› Internet Protocol Version 4, Src: 192.168.70.106, Dst: 10.143.174.177
› Internet Control Message Protocol
```

```
0000  6a be c4 30 c6 cf c4 9d  ed 95 83 04 08 00 45 00   j·0··········E·
0010  00 3c 79 37 00 00 80 01  01 37 c0 a8 46 6a 0a 8f   ·<y7·····7··Fj··
0020  ae b1 08 00 ff 29 00 02  c4 8f 50 00 49 00 4e 00   ·····)····P·I·N·
0030  47 00 20 00 62 00 79 00  20 00 50 00 52 00 54 00   G· ·b·y· ·P·R·T·
0040  47 00 20 00 4e 00 20 20  20 20                     G· ·N·
```

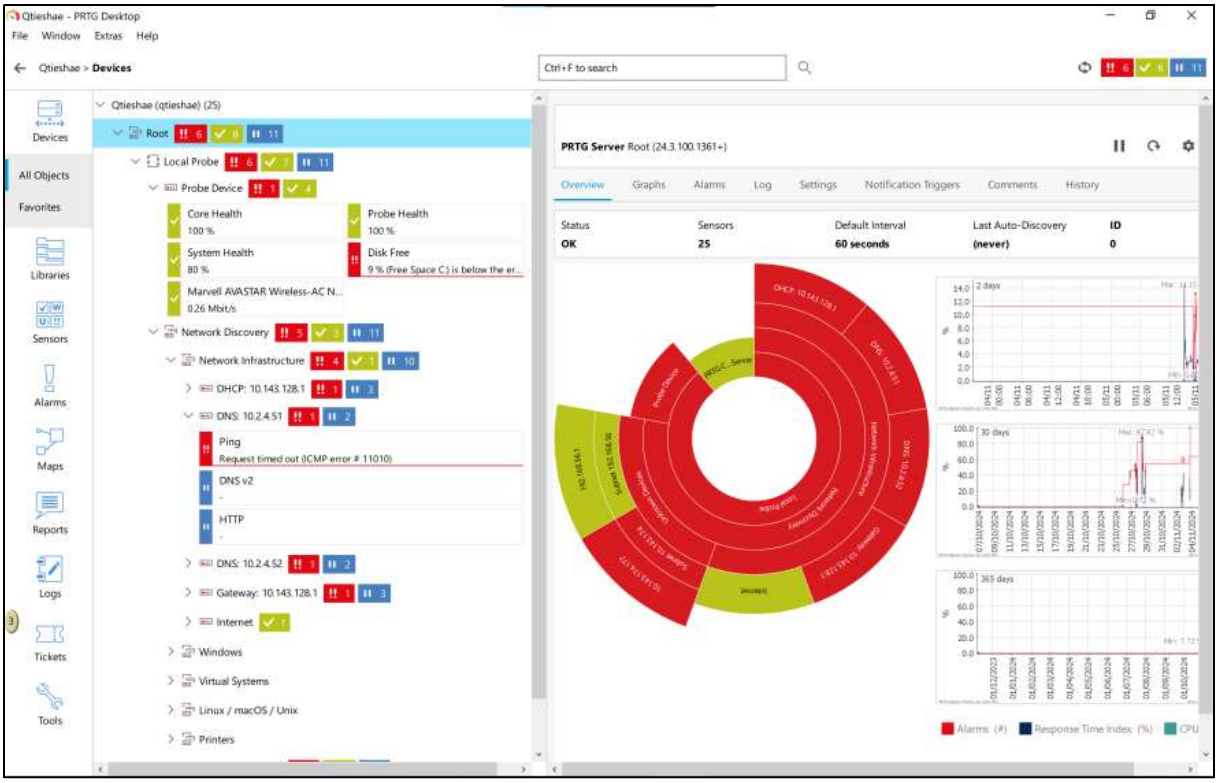
Figure 1. Wireshark Interface

- **Paessler Router Traffic Grapher (PRTG)**

Paessler's network monitoring tool, known as PRTG, serves as an all-inclusive solution for overseeing network performance, designed to help network managers monitor their infrastructure, detect issues, and prevent downtime. It provides real-time visibility into network devices like servers, routers, firewalls, and more, making it suitable for organizations of any size (Nordin, 2021). PRTG's distributed monitoring capabilities offer centralised control of geographically dispersed installations, resulting in seamless performance management across all IT assets. The tool's automatic network discovery makes setup easier by recognising devices and installing sensors automatically. Visual elements such as maps and dashboards provide a clear picture of IT infrastructure, and customisable reports allow managers to follow specific data points, making PRTG an adaptable solution for proactive network monitoring and administration.

Despite its advantages, PRTG has significant drawbacks. Its licensing strategy can be expensive for bigger networks because each monitored item or metric necessitates a new sensor, resulting in potential costs for extensive monitoring requirements. PRTG can potentially become resource-intensive with more sensors, affecting system performance. While the user interface is simple, the tool's broad complexity may be intimidating for newcomers, necessitating time to fully comprehend its potential. Furthermore, while PRTG excels at monitoring and alerting, it lacks the in-depth diagnostic functionality found in specialised troubleshooting tools, thereby limiting its efficacy in fixing specific network issues.

Figure 2 shows one of the interfaces of PRTG. The left side of the graphic shows a list of monitored devices and sensors, such as the main server (PRTG Server Root), local devices, and network discovery tools. Each gadget has a unique health indication (green for healthy, red for concerns). For example, the list indicates that "Probe Device" contains certain cautions about system health and disc space. The middle part provides an overview of the primary server's status, with a colour-coded circle highlighting trouble regions in red (serious issues) and green (no difficulties). The right side shows graphs of response times and CPU consumption over time, which help you visualise how the system is functioning.



Qtieshae (qtieshae) (25)
Root ‼ 6 ✓ 8 ⏸ 11
Local Probe ‼ 6 ✓ 7 ⏸ 11
Probe Device ‼ 1 ✓ 4
Core Health 100 %
Probe Health 100 %
System Health 80 %
Disk Free 9 % (Free Space C:) is below the er...
Marvell AVASTAR Wireless-AC N... 0.26 Mbit/s
Network Discovery ‼ 5 ✓ 3 ⏸ 11
Network Infrastructure ‼ 4 ✓ 1 ⏸ 10
DHCP: 10.143.128.1 ‼ 1 ⏸ 3
DNS: 10.2.4.51 ‼ 1 ⏸ 2
Ping Request timed out (ICMP error # 11010)
DNS v2 -
HTTP -
DNS: 10.2.4.52 ‼ 1 ⏸ 2
Gateway: 10.143.128.1 ‼ 1 ⏸ 3
Internet ✓ 1
Windows
Virtual Systems
Linux / macOS / Unix
Printers

PRTG Server Root (24.3.100.1361+)

Status	Sensors	Default Interval	Last Auto-Discovery	ID
OK	25	60 seconds	(never)	0

Figure 2. PRTG Interface

Results and Discussion

In this comparison, several key criteria will be examined to provide a clearer understanding of each tool's capabilities. The selected criteria include packet capture, filtering, real-time analysis, Graphical User Interface (GUI), protocol dissectors, network visualization, network mapping, bandwidth utilization, and ease of use. Each of these features plays a unique role in effective network monitoring and analysis, contributing to overall efficiency, user-friendliness, and comprehensive insights into network performance.

Firstly, packet capture. This functionality obtains network traffic in real time and converts raw packet data into a comprehensible format. By breaking down each packet, users can gain a detailed insight into the activities that occur at the network layer. Next, filtering. This function enables users to exclude unnecessary information, or "noise," from acquired data, focussing solely on specified network activity or types of communication. This is especially handy for focussing on and resolving specific network issues or protocols without distractions. Furthermore, real-time analysis allows users to monitor live network traffic as it occurs, providing rapid insights. This promptness is critical for recognising and responding to problems as they arise, guaranteeing network stability and security. Moreover, a well-designed Graphical User Interface (GUI) simplifies network analysis for both beginners and specialists. The GUI simplifies work by organising complex data into a user-friendly form, improving efficiency and usability for users of various skill levels.

Other than that, protocol dissectors dissect and interpret numerous protocols, breaking down traffic to highlight crucial data. This gives users a clear, structured perspective of how many protocols interact, which is critical for troubleshooting and analysing network behaviour. Also, the network visualisation tool gives a visual foundation for comprehending network activity in detail. Graphs and visual aids help users understand network flow, traffic surges, and patterns across the whole infrastructure. The next feature is network mapping. Network mapping allows users to view a visual depiction of the complete network, including all linked devices. This map provides a fast overview of the network structure, making it easier to identify connectivity issues or changes in network topology. Additionally, an alerting feature, which is a smart alert warn users when certain thresholds are reached, such as when bandwidth usage surges or latency exceeds a predetermined limit. This proactive alerting prevents minor concerns from worsening by urging timely replies.

In addition, a customisable dashboards feature collects and displays the most critical indicators, offering a quick overview of the network environment. These dashboards can be configured to display certain metrics that are most important to the organisation, allowing for faster decision-making. Aside from that, bandwidth utilisation is a feature that operates by monitoring bandwidth usage. It aids in determining typical and irregular traffic patterns, identifying potential bottlenecks or security issues. This monitoring ensures that the network functions properly and that bandwidth is allocated appropriately. Finally, the ease-of-use feature. This relates to how easy it is to install and integrate the product into an existing infrastructure. A user-friendly technology can be easily embraced by the team, minimising the time and resources required for setup and training.

Table 1 provides a thorough comparison of Wireshark and PRTG, aimed at helping users understand each tool's strengths and limitations to make informed decisions based on their specific network management needs. This comparison evaluates a range of features to highlight each tool's capabilities and offers insight into which tool may be more suitable depending on the use case.

Table 1. Features comparison of each tool

Features	Wireshark	Paessler Router Traffic Grapher (PRTG)
Packet Capture	Recorded network traffic in real time for analysis.	Record network traffic for detailed analysis, troubleshooting, and security investigations.
Filtering	User can filter captured packets using a variety of parameters (e.g., protocol, source/destination IP, port number).	User can filter captured packets depending on a variety of parameters (for example, IP address, port number, or protocol) to isolate specific traffic.
Real-time Analysis	Analyzes network traffic as it is recorded.	Provides immediate insights into network performance, identifying bottlenecks and potential problems as they arise.
Graphical User Interface (GUI)	Offers a user-friendly interface for exploring and analyzing collected packets.	A simple interface for easily navigating, configuring, and monitoring network devices and applications.
Protocol Dissectors	Decodes a variety of network protocols (including HTTP, FTP, SMTP, and SNMP) to display precise packet contents.	Decodes numerous network protocols (e.g., HTTP, FTP, SMTP) to analyze their content and detect potential security concerns or performance issues.
Network Visualization	Depicts network traffic flows and protocol interactions.	Use a visual representation of the network topology to better comprehend complicated network topologies and discover potential risks.
Network mapping	Wireshark can provide insights into network topology through packet analysis, but it cannot automatically build network maps.	Automatically identifies and maps network devices, providing a complete picture of the network infrastructure.
Alerting	Wireshark lacks built-in alerting features. However, user can utilize scripting or third-party tools to generate warnings based	Sends notifications (e.g., email, SMS) when predetermined thresholds are surpassed or certain events occur, allowing for

	on specific traffic patterns or anomalies.	quick response to potential problems.
Customizable Dashboards	Wireshark does not support customizable dashboards. However, user can create bespoke reports and visualizations using its scripting features.	Create personalized dashboards to monitor the most important network data and alarms.
Bandwidth utilization	Wireshark may offer bandwidth consumption information through packet analysis, but it lacks specific bandwidth monitoring capabilities.	Monitors network bandwidth utilization to identify bottlenecks and optimize resource allocation.
Ease of use	Wireshark is a challenging tool to master. It is not as easy to use as certain network monitoring software meant for non-technical users.	The interface is simple and intuitive, requiring little technical knowledge to set up and use.

Conclusion

In comparing Wireshark with Paessler Router Traffic Grapher (PRTG), each tool has unique strengths that cater to different demands and degrees of competence in network monitoring and analysis. Wireshark, recognised for its deep packet inspection and thorough protocol analysis, is an excellent choice for network professionals who need detailed information about traffic patterns and protocol-level interactions. However, Wireshark's steep learning curve and emphasis on packet-level data may provide difficulties for novice or non-technical users looking for a more user-friendly interface for normal network monitoring chores.

PRTG, on the other hand, offers a more comprehensive approach to network monitoring, thanks to its user-friendly interface, automated network mapping, customisable dashboards, and real-time notifications. Its simplicity makes it suitable for both beginners and IT pros, providing a seamless learning experience while still supplying the critical capabilities required for comprehensive network supervision. PRTG's ability to produce network maps automatically, monitor bandwidth, and send notifications when thresholds are exceeded makes it an adaptable tool for both day-to-day monitoring and proactive issue solving.

To summarise, PRTG is the suggested tool for users of all skill levels who require a comprehensive and easy-to-use network monitoring solution. It provides a balanced mix of automated warnings, customisable dashboards, and a variety of monitoring sensors, making it appropriate for both large and small networks. Its ease of use, paired with advanced features such as remote monitoring and compatibility with various network protocols, ensures that it fits the demands of professionals while also being accessible to beginners. Wireshark, on the other hand, is invaluable for users who need very comprehensive packet-level analysis and is extremely successful in circumstances requiring in-depth troubleshooting or security investigations.

Combining both tools as complimentary resources allows for a more comprehensive approach to network monitoring, with users benefiting from both broad network insights from PRTG and fine-grained packet analysis from Wireshark. This technique enables network administrators to keep a broad perspective of network health and identify concerns quickly, while also having the option to go deeper as necessary. Furthermore, using both tools increases the flexibility to address a variety of network monitoring requirements, such as routine performance tracking, anomaly detection, and root-cause analysis of network issues, making them a perfect combination for a wide range of network management and troubleshooting scenarios.

Acknowledgements

The authors would like to express their sincere appreciation to the members of the School of Computing for their essential contributions to this study. This work was carried out as part of the System and Network Security Project, with the support of Universiti Utara Malaysia.

References

Alip, N., Fitri, I., & Nathasia, N. D. (2018). Network monitoring system data radar penerbangan berbasis PRTG dan ADSB. *JOINTECS (Journal of Information Technology and Computer Science), 3*(3). https://doi.org/10.31328/jointecs.v3i3.818

Banerjee, U., Vashishtha, A., & Saxena, M. (2010). Evaluation of the capabilities of Wireshark as a tool for intrusion detection. *International Journal of Computer Applications, 6*(7), 1–5. https://doi.org/10.5120/1092-1427

Botchway, T. (2022). Analysing network information and protocol using Wireshark. *Advances in Multidisciplinary and Scientific Research Journal Publication, 1*(1), 203–208. https://doi.org/10.22624/aims/crp-bk3-p33

Clegg, R. G., Withall, M. S., Moore, A. W., Phillips, I. W., Parish, D. J., Rio, M., Landa, R., Haddadi, H., Kyriakopoulos, K. G., Auge, J., Clayton, R., & Salmon, D. (2009). Challenges in the capture and dissemination of measurements from high-speed networks. *IET Communications, 3*(6), 953–957. https://doi.org/10.1049/iet-com.2008.0068

Dodiya, B., & Singh, U. K. (2022). Malicious traffic analysis using Wireshark by collection of indicators of compromise. *International Journal of Computer Applications, 183*(53), 1–6. https://doi.org/10.5120/ijca2022921876

Jain, G., & Anubha. (2021). Application of SNORT and Wireshark in network traffic analysis. *IOP Conference Series: Materials Science and Engineering, 1119*(1), 012007. https://doi.org/10.1088/1757-899X/1119/1/012007

Kim, J., & Sim, A. (2019). A new approach to multivariate network traffic analysis. *Journal of Computer Science and Technology, 34*(2), 388–402. https://doi.org/10.1007/s11390-019-1915-y

Mala, B., Agrawal, S., Sharma, A., & Kaur, R. (2023). Exploring Wireshark for network traffic analysis. *International Journal for Research in Multidisciplinary Fields, 5*(6). https://www.ijfmr.com/papers/2023/6/8876.pdf

Musa, A., Abubakar, A., Gimba, U. A., & Rasheed, R. A. (2019). An investigation into peer-to-peer network security using Wireshark. *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, 1–6. https://doi.org/10.1109/ICECCO48375.2019.9043236

Ndatinya, V., Xiao, Z., Manepalli, V. R., Meng, K., & Xiao, Y. (2015). Network forensics analysis using Wireshark. *International Journal of Security and Networks, 10*(2), 91–100. https://doi.org/10.1504/IJSN.2015.070421

Nordin, M. (2021). *Implementing a monitoring system using PRTG* [Bachelor's thesis, Häme University of Applied Sciences]. Theseus. https://www.theseus.fi/bitstream/handle/10024/504829/Nordin_Mats.pdf?sequence=2

Pavithirakini, S., Bandara, D., Gunawardhana, C., Perera, K., Abeyrathne, B., & Dhammearatchi, D. (2016). Improve the capabilities of Wireshark as a tool for intrusion detection in DOS attacks. *International Journal of Scientific and Research Publications, 6*(4), 378–382. https://www.ijsrp.org/research-paper-0416/ijsrp-p5259.pdf

Praful, S., Sharma, S., & Kumar. (2017). Analysis of network traffic by using packet sniffing tool: Wireshark. *International Journal of Advance Research, Ideas and Innovations in Technology, 3*(6). https://www.ijariit.com/manuscripts/v3i6/V3I6-1369.pdf

Qureshi, S., Li, J., Akhtar, F., Tunio, S., Khand, Z. H., & Wajahat, A. (2021). Analysis of challenges in modern network forensic framework. *Security and Communication Networks, 2021*, Article 8871230. https://doi.org/10.1155/2021/8871230

Hashim, S. R., Enad, R. A., Al-Khafagi, M. A., & Abdalhameed, N. K. (2023). The facilities of detection by using a tool of Wireshark. *Indonesian Journal of Electrical Engineering and Computer Science, 31*(1), 329–336. https://doi.org/10.11591/ijeecs.v31.i1.pp329-336

Rachman, D. A., Muhyidin, Y., & Sunandar, M. A. (2023). Analysis quality of service of internet network fiber to the home service PT. XYZ using Wireshark. *Jurnal Informatika dan Teknik Elektro Terapan, 11*(3s1). https://doi.org/10.23960/jitet.v11i3s1.3436

So-In, C. (2006). A survey of network traffic monitoring and analysis tools. *ResearchGate.* https://www.researchgate.net/publication/241752391_A_Survey_of_Network_Traffic_Monitoring_and_Analysis_Tools

Soepeno, R. (2023). Wireshark: An effective tool for network analysis. *ResearchGate.* https://doi.org/10.13140/RG.2.2.34444.69769

Sulicdio, J., Kalsum, T. U., & Arliando, Y. (2022). Comparative analysis of Wireshark and Windump software in network security monitoring. *Jurnal Media Computer Science, 1*(1). https://doi.org/10.37676/jmcs.v1i1.1901

Tuli, R. (2023). Analyzing network performance parameters using Wireshark. *International Journal of Network Security & Its Applications, 15*(1), 1–13. https://doi.org/10.5121/ijnsa.2023.15101