RNMS-PR: Remote Network Monitoring System with Promstack on Raspberry-PI

A. Abdallah, I. Zouari, M. Abdelwahab, M. Amin, and M. Kechaou

***Abstract*— As a response to the increasing challenges of the growing density and complexity of data traffic, this project proposes a network monitoring system that combines Raspberry Pi with Prometheus, Grafana, and ZeroTier. The traditional networks' monitoring systems are inconvenienced by the size and sophistication of modern networks. In our approach, we employ Raspberry Pi to remotely detect and collect data, Prometheus for data collection, and Grafana to visualize data with the help of a custom dashboard. As Prometheus is an open-source monitoring and alerting toolkit for gathering and storing metrics as time series data, Grafana could be necessary because it is visualization and analysis software for the network performance metrics. As part of the network monitoring system, we intend to include ZeroTier which will provide security and accessibility in terms of setting up a private network for the administrators. The ZeroTier IP network can be set up such that only the devices that are authorized can access the monitoring system that is on the Raspberry Pi. The project involves making the monitoring isolated and protected from external threats by combining ZeroTier with Raspberry Pi, Prometheus, and Grafana. Our application was tested in a real-world dense network setting, showing precise monitoring and a consolidated view of the important network metrics. The present research aims to improve network traffic monitoring techniques by introducing more efficient and accurate methods which is a substantial contribution to the field and can open the door for potential future applications and developments. This project proposes a powerful and comprehensive solution for traffic monitoring and management that is flexible enough to cope with an ever-changing network environment by leveraging the strengths of Raspberry Pi, Prometheus, Grafana, and ZeroTier.**

I. INTRODUCTION

IN a modern network environment with growing intricacy and tons of traffic, we face even tougher problems. As the rapid increase of online service is coupled with cloud computing and IoT devices, it has never been more necessary to deploy strong network traffic monitoring solutions. Addressing these issues poses a great challenge and the project has an elegant solution of a cutting-edge method of real-time monitoring based on the utilization of microcomputers and software techniques for visualization and monitoring. Through Raspberry Pi serving as the foundation of the system, the plan is to offer an affordable, customizable, and multifunctional platform that both fulfills the present needs of the networks and determines a new level for future efficient and mobile monitoring. The combination of the ZeroTier technology with the Raspberry Pi-hosted website not only delivers the system security and accessibility jump but also ensures connectivity across the network where the data are encrypted from the end to the end. This configuration on the other hand is not only about easy access to the hosted website, but also it acts as a secure platform that is responsible for real-time monitoring and control of network traffic. Real-time monitoring and control systems are conducive to the development of modern networks concerning the evolution of remote access solutions.

A. Background

In the domain of network monitoring, previous projects have recognized the importance of effective and trustworthy methods to deal with the problems modern networks are facing. Tools such as Cacti, Nagios Core, Icinga2, and Zabbix played a crucial role in collecting performance data and ensuring network health with features like SNMP support and customizable alerting mechanisms providing necessary resources for the network administrators. Network monitoring software assumes such an important function in performance enhancing, reducing downtime, improving security, and streamlining operations thus allowing organizations to achieve cost savings, ensure compliance, and have a complete view of the network's status and usage. The flexibility of network monitoring solutions, ranging from open-source systems to enterprise platforms, empowers organizations to adjust their approach to meet specific needs and integration requirements and evoke this project to explore new approaches to address the challenges of network monitoring dynamically.

B. Problem Statement

Concerning flow traffic monitoring, the prevalent methods run into several issues in the space of scaling and complexity of existing networks. Traditional monitoring mechanisms have scalability and complexity complexities that make it hard for them to exactly measure the indispensable network metrics. This gap depicts the demand for a new approach that will interlink Raspberry Pi, Prometheus, Grafana, and ZeroTier just to have the perfect network monitoring systems, precision, accuracy, and security. Through the use of advanced functions of Raspberry Pi in terms of remote capabilities for data collection, Prometheus for its data collection capacity, Grafana for its visualization capabilities, and security provided by ZeroTier, there is a need to build up a paramount solution that can act as a stepping stone in tackling the ever-advancing needs of network monitoring in this technologically driven society. The research is aimed at filling this gap by suggesting a new methodology that covers not just the constraints of traditional monitoring procedures but also establishes a new standard for high-precision, effective, and safe network traffic monitoring in a dense network scene. This research attempts to give prominence to the discipline by suggesting more efficient and accurate ways for data transferring, hence preparing the ground for tomorrow's enhancement and applications in network monitoring technology.

C. Project Objectives

The objectives of remote network monitoring aim to provide a transversal and detailed monitoring solution for network managers and users. The major objectives are to develop and integrate into the network monitoring framework a system with good scalability and good handling of a large quantity of network data to ensure real-time monitoring features for proactive network management. The implementation of this system should run based on Prometheus and Grafana, as they are useful to have a customizable monitoring platform so that the users create dashboards for their specific needs and help monitor various network metrics, such as network traffic.

Furthermore, the platform needs to be integrated with ZeroTier for secure and efficient administration of the network, involving the creation of virtual networks and the management of devices and services on the network. Additionally, an Apache server will be set up on the Raspberry Pi to host the website, which is the platform of the system to facilitate access to the Grafana dashboard, which will have a user-friendly interface, thus allowing users to view and interact with the monitoring data.

Moreover, the user interface of the system should be designed to let the users to access monitoring data using a customizable dashboard containing warnings and notifications to make the network management easy. In this regard, it is also mandatory to develop a security plan together with it. An example is developing strong authentication and authorization mechanisms that can prevent unauthorized access and ensure that the data obtained will be safe as a result. On top of that, it should be multipurpose, let the incorporation of additional monitoring sensors and technological advancements in and cover the future demands.

Finally, the system should be inexpensive and energy efficient. The Raspberry Pi and other low consumption and affordable components should be able to cope with a wide range of environments requirements.

II. Literature Review

A. Existing Work

The deployment of Prometheus with the inclusion of Node Exporter, SNMP Exporter, and Black-Box on the top of the Raspberry Pi with ZeroTier for remote access marks a big breakthrough in the remote network monitoring systems' world. Such a method has several major advantages as compared to the monitoring solutions that were used before. The approach offers companies a tool that makes it possible for them to have effective network observability and raises the level of troubleshooting they do. Prometheus, which is an open-source tool that is very well performing in collecting metrics, storing them, and also querying them, has gained huge popularity among people [11]. Its data collection function is made possible by continuous data collection technology, and TSDB PromQL allows for operational data analysis and visualization. The Prometheus Blackbox Exporter lets us not only visualize what is happening on the server side but also check the endpoints employing protocols such as HTTP, TCP, and ICMP to gain information about network connectivity, latency, and packet loss [11].

In this project, Node Exporter, SNMP Exporter, and BlackBox facilitate the collection of performance data from network devices, making it possible to expand monitoring to include important control parts, including firewalls and switches. This approach makes it different from traditional systems, such as Zabbix or LibreNMS, that generally do not provide such tools for customization and data analysis. The integration of the system on the Raspberry Pi layer sets a mobile and highly versatile interactive layer onto the monitoring system allowing it to be used in different conditions effectively. On the other hand, in the case of fusion, no separate devices or central monitoring servers are required, unlike conventional merges, which can be very resource-consuming and less flexible when it comes to handling situations. Virtual network ZeroTier will make project access more secure and remote, and in addition, remote access to a Raspberry Pi-based monitoring system will offer several significant advantages to the project by allowing for real-time distributed network monitoring [12]. This component distinguishes this project from all the previous solutions that may have required complicated VPN installations or other remote access procedures for remote users. Among others, the project uses an Apache server on the Raspberry Pi to deliver the website that is the gateway to the monitoring system. This website, like the monitoring solution itself, is only granted access to devices connected to the private ZeroTier network, thus consolidating security and accessibility.

Unlike in the past with network monitoring projects, the set-up, which includes Prometheus, Node Exporter, SNMP Exporter, and Black Box using Raspberry Pi with ZeroTier, provides numerous benefits over the traditional set-up of monitoring the network. Whereas the traditional approach may achieve objectives such as agent-based data collection and built-in alert systems, the new approach outperforms in terms of customization, data analysis tools, and a monitoring solution capable of handling a dynamic network environment [13]. In addition, the project's capability of monitoring complex systems that have multiple dependencies, as well as the high configurability and customization options that come with it, make it a pioneering solution in the hitherto challenging environment of monitoring networks. Applying modern approaches to monitoring systems allows organizations to raise the level of their monitoring, gain deeper knowledge about process performance, and maintain stability and visibility.

In summary, Whereas the monitoring domain endlessly changes, using new tools becomes a crucial driving force of the future, so these tools enable organizations to reach significant system efficiency, reliability, and observability. This project provides an advanced solution for remote network monitoring through the combination of Prometheus, Node Exporter, SNMP Exporter, and Black-Box on Raspberry Pi with the addition of ZeroTier for connectionless and IP-independent remote access, along with the use of an Apache server to host a secure monitoring website.

B. Contributions and Efforts

The project makes a contribution to network monitoring and metrics by introducing a system for data collection that actively uses Prometheus and Grafana as analytical tools. This approach led to the discovery of new ideas based on the gathering and analysis of information that was not available before. The deployment of the newest microcomputer technology, ruling out the Raspberry Pi method, proves the practicality of monitoring approaches for networked environments. In the process of thorough real-life scenario trials, the system has been evaluated for possible efficiency and accuracy, ensuring a reliable system stakeholders can use to get precise network metrics. During the project, the deep research revealed the important elements of the network operation and provided necessary knowledge about the network dynamics and improvement strategies.

The project team has shown that they have a diverse skill set, utilizing it during the research and deployment phases. The experience with Bash scripting proved the importance of proficiency in automating system management and monitoring, where SNMP was utilized as one of the network protocols. As evidenced by the ability to configure and manage an Apache server on the Raspberry Pi, the knowledge of server administration provided uninterrupted operations and maintained data integrity. The fact, that the team could install and set up the Raspberry Pi boards for monitoring purposes and even integrate them with the ZeroTier for secure remote access, showed a high level of technical skills. Along with the troubleshooting abilities and system management backed up the competency in keeping up the system's reliability and availability. Of note was a proper allocation of time for planning and project execution, which portrayed the team's dedication to project success. The collective teamwork utilized during the process of project development and evaluation proved to be the key factor in reaching the project objectives and achieving the project's success.

Besides, the project team demonstrated a high level of problem-solving capacity in overcoming the difficulties that came with the remote monitoring of the network. The conclusion of the Prometheus Blackbox Exporter integration, which is employed in corresponding protocols, has proved the team's ability to suit different monitoring requirements. The team demonstrated knowledge of different monitoring techniques and a great ability to expand the system monitoring tools, using different exporter modules for retrieving performance data from network devices. On top of that, the team created dashboards and reports that gave network administrators a full view of their infrastructure, which was made possible through the use of Grafana.

By the end of the project, it can be safely concluded that the network monitoring and metric visualization fields have been greatly enhanced. The Raspberry Pi system, complete with Prometheus, Grafana, and ZeroTier technologies, shows the group's competency in technical issues, problem-solving, and delivering high-quality and versatile monitoring solutions. The accomplishment of the project proves that the team had all the right skills, excellent teamwork, and an unflinching dedication that led to the establishment of new approaches to network monitoring practices.

III. Methodology

A. Research Design

This study employs a descriptive research design. To assess the performance and effectiveness of network monitoring done by our remote system. The network aspects evaluated are bandwidth, throughput, latency, Round Trip Time (RTT), packet loss, Signal-to-noise ratio (SNR), and Bit Rate (BR).

B. Data Collection

Primary data were collected through test assessments done on the Raspberry PI. Test assessments consisted of pinging, accessing the device, gathering, and exporting metrics of the network's performance from the Raspberry PI. These test assessments depend on the exporters used in this project, Node Exporter, BlackBox Exporter, and SNMP Exporter.

C. Data Analysis

Quantitative data analysis was conducted using Prometheus and Grafana, to gather, classify, analyze, and visualize the data. Descriptive statistics and plots were calculated to summarize the performance of the network. Quantitative analysis is the best approach, since the aim is to continuously measure the metrics and evaluate them in terms of accuracy and performance. The interesting metrics are: bandwidth, throughput, latency, Round Trip Time (RTT), packet loss, Signal-to-noise ratio (SNR), and Bit Rate (BR).

D. Experimental Setup

The experimental trials were conducted on different Virtual Machines on different devices and on the Raspberry PI, all of which were connected to different types of networks, including mobile hotspots, home wireless networks, and free public networks. These different environments were chosen because of their different traffic, number of users, and protocols used. The virtual machines deal differently with the network as they are concerned with security, whereas different devices have different usage and connectivity to the network. The main aim of the experimental trials is to assess the effectiveness of the developed framework in different scenarios.

E. Software Setup

Multiple software tools were used in the development of the Promstack for the project including Prometheus, Grafana, SNMP Exporter, BlackBox Exporter, Node Exporter, and ZeroTier all setup in the developed and well commented bash scripts.

1) Prometheus

Prometheus is a monitoring tool used to monitor highly dynamic container environments and servers with the application directly installed in them [2]. It became the main option for container and microservices infrastructure. It is written in Go programming language. What is being monitored by Prometheus is called the "Target" and configured in the Prometheus configuration file and the gathered data are called "Metrics", which will be saved in the Prometheus server. Metrics have unique characteristics such as being time series data, which means that all the gathered data are saved with time stamps. Prometheus has different metric types, Counter, which represents a cumulative metric, Gauge, which represents the current value of a metric, and Histogram, which represents how values are distributed over time. The Prometheus server works in three steps, Data Retrieval, Data Storage, and PromQL Queries. Data retrieval is done by a pull system, by pulling the metrics from HTTP endpoints and pushing these metrics with the timestamps to do the data storage. Data storage is done by saving the data with the timestamps in a dedicated database. PromQL queries are done to convert the data into an appropriate format so that we can query the data externally using Prometheus web UI or Grafana. Additionally, Prometheus Alert Manager can send alerts by using the metrics from Time Series Data Base (TSDB). These alerts can be used for troubleshooting and identifying the problems that came about and ensuring that they do not happen again.

2) Exporters

Exporter is a software that can be installed and configured at an application or a device to be monitored. Prometheus server will then make a request to the exporter, and the exporter will retrieve the specific data from target and build the receiving format as requested by the server. Data gathered will then be fed back to Prometheus server where it will be stored. Prometheus supports many exporters that are developed by the Prometheus developing team and other Third-party developing teams.

3) Node Exporter

Node exporter is designed to collect data from Unix environments [8]. It exposes hardware and software metrics from the target machine such as CPU, memory, and disk space metrics. Those metrics are made accessible through a URL that aligns with Prometheus standards. With a wide range library of supported applications, the Node Exporter enables the exportation and transformation of metrics from third-party sources into Prometheus-compatible metrics. Node exporter is used to give an insight into the node's health through those metrics.

4) Blackbox Exporter

The Blackbox exporter is one of many ways of providing Prometheus with metric data from systems whose internal mechanisms cannot be accessed by Prometheus scraper. It will perform the HTTP, TCP, ICMP and then write the response times as well as the status to a file. It is configurable and customizable because users can define endpoints such as HTTP endpoints with specific headers, TCP endpoints with custom payloads and ICMP endpoints with actual payload sizes. Moreover, users can specify timeouts and authentication options for each endpoint. It can work in conjunction with other sensors to enable and monitor complex systems. It can also collect metrics on health and response times of other endpoints with the help of protocols such as HTTP, HTTPS, ICMP, DNS, and TCP.

5) SNMP Exporter

SNMP exporter is used for tracking devices that have the SNMP capability on, for instance, switches and routers. The Exporter and Generator components are two main elements contained within it. In this case, Exporter, plays the role of an agent, which can get the statistics from devices managed through SNMP. It is, in fact, the network management system (NMS). This is how SNMP data are exposed to Prometheus. Generator, this is the component that creates the configurations for the exporter by mapping SNMP object identifiers (OIDs) to counters and gauges that can be understood by Prometheus.

6) Grafana

Grafana is an open-source software that specializes in data analysis and visualization all through custom-made dashboards [3]. Grafana is a kind of system that includes various connectors that can be connected to different tools, databases, servers, and sources. It is a very powerful instrument that can experience and grasp complicated data sets and metrics in a very comprehensive view and contentment. Grafana contains a broad range of community-developing plug-ins and integrations that permit users and developers to customize Grafana to their own needs..

7) ZeroTier

ZeroTier is the software that allows the creation of virtual private networks for multiple devices to connect irrespective of the fact that those devices are connected to some other network [4]. ZeroTier is based on an encryption channel that creates an encrypted Local Area Network (LAN), which devices logically connect to. When a new device wants to connect to a virtual network the owner of the network must authorize this attempt of connection, after the access grant ZeroTier gives the new device a unique nodeID and the networkID. Additionally, ZeroTier creates a VL1 network. VL1 is the peer-to-peer network layer in ZeroTier that establishes direct, encrypted connections between nodes using globally unique 40-bit ZeroTier addresses assigned by the root servers, which act as a directory service to resolve these addresses to IP addresses, like how DNS resolves domain names to IP addresses. The VL1 layer handles the low-level networking tasks, while the higher-level VL2 layer provides Ethernet virtualization and software-defined networking features on top of the VL1 connections.

F. *Validity and Reliability*

To assess the performance of the project, we had the equivalent network metrics generated using other systems to be compared with the results of our system. A rigorous testing process was carried out for both the systems – be it the traditional or our systems – in different environments to verify the correlation between the results of both systems.

G. *Ethical Considerations*

Ethical approval for the project testing was obtained from the owners of private networks before data collection. Informed consent was obtained from all participants with measures to protect their privacy.

H. *Limitations*

Limitations of the project include the potential for different results of real-world performance due to external factors, such as: the number of users, the distance between the device and the router, and type of devices connected to the network. Another limitation of the project is the scalability, future research and enhancements are required to be able to deploy the system on very large and dense networks.

IV. Implementation

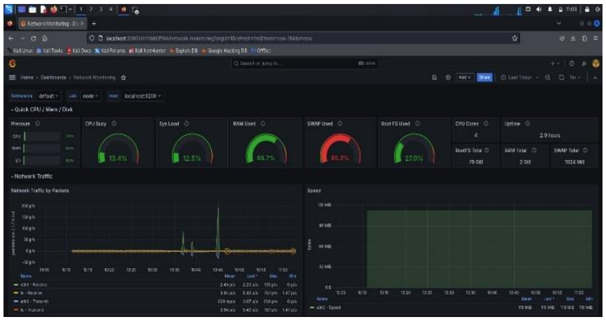
Implementation of the proposed monitoring system involved the development of a prototype that involved hardware components, software frameworks, and custom-made configurations for the attainment of the objectives in the Project Objectives section above.

A. *Hardware Setup*

The hardware setup consisted of a Raspberry Pi 4 Model B microcomputer as the primary monitoring device. The Raspberry Pi possessed all necessary peripherals including an Ethernet adapter, the power supply unit, the heatsink, the cover case, the HDMI converter, and storage for its operation as a standalone remote monitoring unit. Adopting Raspberry PI as a means of building a network monitoring system has several benefits, for example, low cost, minimum power consumption, very high customizability, very high versatility, very high compatibility with different ecosystems and devices, and improved mobility. Additionally, Kali Linux was installed and configured on the Raspberry Pi as its operating system. SSH was set up on the Raspberry Pi to allow secure remote access to the Raspberry Pi Operating System. Additionally, ZeroTier software was set up to create a virtual private network that an authorized user can connect to from anywhere in the world and access the Raspberry Pi, which allowed controlled remote access to the Raspberry Pi.

B. *Software framework*

Promstack is the software framework that is used in order to implement the project. Promstack framework uses Prometheus, Grafana, and other Exporters, for example, SNMP, BlackBox, and node exporters. Prometheus is configurable and employed for the following functions: data harvesting, collection, and archiving. As well, Grafana was set up with Prometheus as a main data source and was utilized for data visualization with the use of Prometheus. The graphical user interface was designed to enable customers to customize the dashboard and, at the same time, facilitate easy analysis of the data. In addition to this, SNMP, BlackBox, and Node Exporters were used to produce metrics. BlackBox, Node, and SNMP Exporters were used to get metrics of the network including Throughput, Packet Metrics, Error Metrics, Drop Metrics, Network Status Metrics, Socket Statistics, and TCP Connection States. In Grafana various metrics are employed to visualize different sides of the network health and performance. Throughput metrics like node_network_receive_bytes_total and node_network_transmit_bytes_total keep a tab on the total bytes received and transmitted. Packet metrics node_network_receive_packets_total and node_network_transmit_packets_total, where total packets received and transmitted are monitored respectively. Error statistics, drop statistics, and network status metrics reveal network health and issues, On the other hand, statistics about sockets like node_sockstat_sockets_used and node_sockstat_TCP_inuse, bring information about socket usage. TCP connection states, as noted by [node_tcp_connection_states], specify the amount of TCP connections in different states. These statistics altogether provide detailed monitoring of the whole network functioning and situation inside Grafana. More than a JSON file has been created for our Grafana dashboard configurations. The ZeroTier installation process starts with visiting the official website, creating an account, reading through instructions, and lastly, setting up a private network. Furthermore, a unique network ID will be automatically assigned for the created network and a list of possible private IP address ranges will be available to choose from. Configuration files, Scripts, and any other used software-related files are available in detail within a guide pdf in the submitted CODE/DEMO report. Moreover, an HTML webpage was set up as a graphical user interface (GUI), and it includes links to the Grafana Dashboard, GitHub project page, and Technical Report as the project's documentation. Below is a picture of the developed Grafana Dashboard.



C. Integration and testing

After developing the Promstack, it was installed and configured on the Raspberry Pi's Kali Linux. The written scripts were run on the Raspberry Pi to complete the integration process and automate the monitoring process.

Different test scenarios were conducted under different network conditions to evaluate the efficiency and accuracy of the developed monitoring system. Different networks have different numbers of users connected, different traffic loads, different traffic patterns, different protocols, and different configurations of the devices connected.

Validating the outcomes of the monitoring system was done by comparing the collected metrics with the metrics collected from manual measurements and traditional monitoring tools and systems. The faults and inconsistencies were identified and solved for better quality and precision of the created monitoring system.

D. Deployment and Optimization

Following successful installation, integration, and testing, the network was tested out in several practical network environments, including mobile hotspots, private networks, and public networks. The configuration was going to include features like the scrape interval of the exporters used showing which metrics were needed for this environment and optimizing the system.

To succeed, continuous monitoring, upgrading, and optimization of the improved system was done after the deployment of the monitoring system to be able to identify and correct upcoming problems and errors.

E. Documentation

Extensive preparations were made to create guides for system startup and operation, troubleshooting, and anticipated future improvements. In addition, both the installation instructions and the information on how the operating system works are available on GitHub and in the submitted folders.

V. Results

The project's outcomes include the creation of a website as a platform for the system that is hosted on an Apache2 server on a Raspberry Pi. The system uses Prometheus' functionalities and protocols to gather and retrieve metrics from the Raspberry Pi node. For a new user or to be able to see results installation of a zerotier is required. After obtaining a grant from the owner an IP will be distributed, then and only then access to the website is granted through http://10.147.19.95 which is the IP of the Server hosted on the Raspberry Pi. The website of the system includes a link to a Grafana dashboard, which allows users to real-time monitor metrics in graphical form. The graphs presented on the Grafana dashboard are the result of Prometheus numerous functionalities and protocols for gathering and analyzing information on the Raspberry Pi node. These metrics cover a wide range of network-related metrics, including network traffic, faults, drops, status, socket statistics, TCP connection statuses, and more. The system's versatility enables the study of findings from a variety of viewpoints, depending on the individual application and exporter configurations used in the network monitoring process.

A. Metrics

The metrics obtained from the system are as follows.

1) Network Traffic

The network traffic metrics gathered by the Prometheus Node Exporter give a detailed picture of data flow and packet processing in the network architecture. The *node_network_receive_bytes_total* and *node_network_transmit_bytes_total* metrics show the total number of bytes received and transmitted across network interfaces, demonstrating the network's capacity to manage the present data load efficiently. Packet-related metrics, such as *node_network_receive_packets_total* and *node_network_transmit_packets_total*, provide insight into overall network traffic patterns and assist in identifying possible packet processing difficulties.

2) Error and Drop

Error and drop metrics are crucial for assessing the reliability and health of network connections. The *node_network_receive_errs_total* and *node_network_transmit_errs_total* metrics capture the total number of receive and transmit errors encountered by network interfaces, assisting in the discovery and resolution of data corruption, network congestion, and interface-level problems. The *node_network_receive_drop_total* and *node_network_transmit_drop_total* metrics provide information on the total number of dropped packets received and sent, enabling network issues to be recognized and remedied.

3) Network Status

The network status metrics, such as *node_network_up*, *node_nf_conntrack_entries*, and *node_nf_conntrack_entries_limit*, offer valuable information about the overall connectivity and state of the network. The *node_network_up* statistic reflects the operational status of the network

4) Interfaces

Interfaces, which aids in promptly discovering and resolving connection issues. The *node_nf_conntrack_entries* and *node_nf_conntrack_entries_limit* metrics offer information on the use of the *conntrack* table, which is in charge of tracking network connections, enabling for the discovery of possible connection tracking issues.

5) Sockets Status

The socket-related metrics, such as *node_sockstat_sockets_used*, *node_sockstat_TCP_inuse*, and *node_sockstat_UDP_inuse*, give information on the network's socket usage and resource consumption. Monitoring the overall number of sockets utilized, as well as the particular use of TCP and UDP sockets, can help identify possible difficulties caused by socket exhaustion or uneven resource consumption, allowing for network performance optimization and efficient connection handling.

6) TCP state

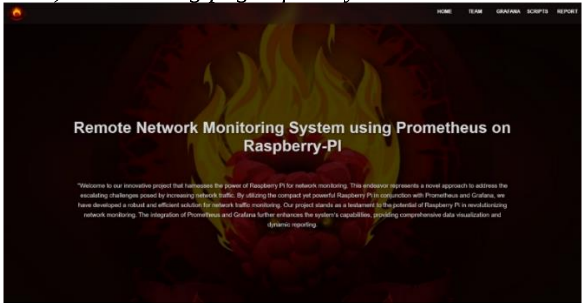
The *node_tcp_connection_states* metric gives a thorough overview of the different states of TCP connections in the network. Analyzing this measure can aid in understanding overall TCP connection dynamics, recognizing potential connection-related issues, and optimizing network resources as needed.

To conclude, the complete set of metrics gathered by the Prometheus Node Exporter, SNMP Exporter, and Blackbox Exporter provides a detailed and analytical perspective of the network's health and performance. By monitoring critical metrics such as network traffic, errors, dropouts, status, socket utilization, and TCP connection statuses, the network monitoring system may efficiently identify and resolve possible issues, enhance network performance, and assure overall infrastructure dependability.

The study of these indicators has shown that the established network monitoring system is successful in delivering significant insights and enabling informed decision-making for network administration and enhancement

B. User Interface

1) Welcoming page of the system

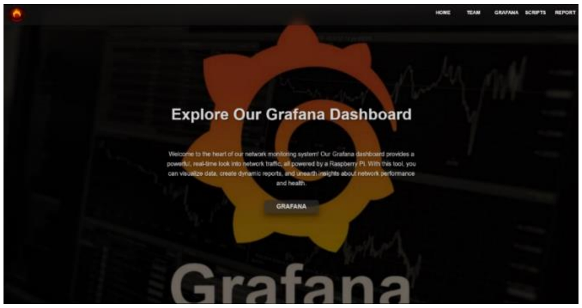


Remote Network Monitoring System using Prometheus on Raspberry-PI

"Welcome to our innovative project that harnesses the power of Raspberry Pi for network monitoring. This endeavor represents a novel approach to address the escalating challenges posed by increasing network traffic. By utilizing the compact yet powerful Raspberry Pi in conjunction with Prometheus and Grafana, we have developed a robust and efficient solution for network traffic monitoring. Our project stands as a testament to the potential of Raspberry Pi in revolutionizing network monitoring. The integration of Prometheus and Grafana further enhances the system's capabilities, providing comprehensive data visualization and dynamic reporting.

2) *A link to the Grafana dashboard*

A link to the Grafana dashboard from the system:

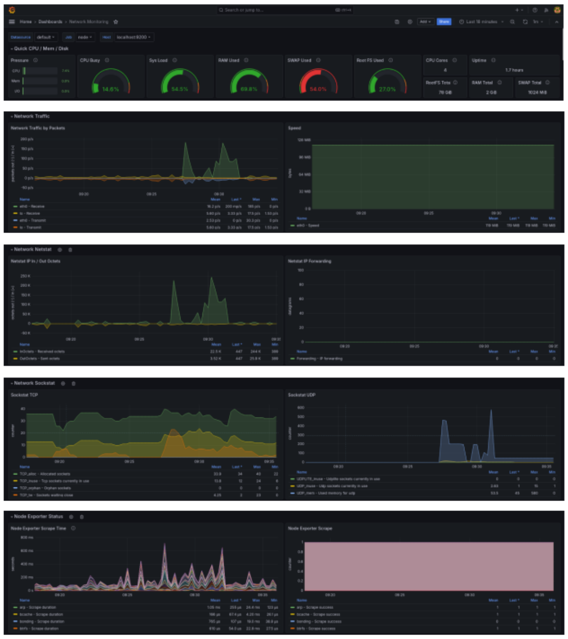


Explore Our Grafana Dashboard

Welcome to the heart of our network monitoring system! Our Grafana dashboard provides a powerful, real-time look into network traffic, all powered by a Raspberry Pi. With this tool, you can visualize data, create dynamic reports, and unearth insights about network performance and health.

3) *Grafana Dashboard*

Grafana dashboard contains real-time monitored metrics:



VI. Individual Contributions

A. *Ahmad Abdalla*

Ahmad designed the maintenance phase of the project and obtained all the required parts needed for fabrication and assembly of the device. He was a crucial member of the team the whole time. Along the way, he rolled up his sleeves and worked on different parts, like writing the software frameworks and building the HTML page of the project. During the execution phase, Ahmad came up with other strategies including re-configuring the software by revising the software's architecture and rectifying numerous problems we encountered. Ahmed's input was great to achieve our project goals and also changing as new challenges are unfolding.

B. *Iheb Zouari*

From the initial project plan, Iheb joined in the network configuration, debugging, and testing phases. He implemented the proposed plan by following all the recommendations for the network configuration, as well as configuring and implementing JSON file for Grafana dashboard. With the project in progress, Iheb tweaked the program to eliminate any problems that might happen. Additionally, Iheb responded to evolving demands and further enhanced the application's performance. Further, he was also involved in writing the report, guidelines for implementation, and the analysis of results. His devotion to detail and an active approach was a great factor for the successful outcome of the project. Furthermore, it was Iheb who took part in the making of the framework, particularly in the generation of the scripts needed for automation process.

C. *Moustafa Abdelwahab*

Following the project plan that was developed at the beginning, Moustafa had a crucial role in the research phase, software development, and testing phases. He eagerly followed the recommended structure, applying the known best practices for researching, software development, and testing during the project. In the end, Moustafa installed and configured both the Blackbox and SNMP exporters. Besides, he also collaborated on the documentation about Prometheus and its exporters. With the inspiration of his team members, Moustafa brought necessary changes to the process of developing the software framework which contributed to the improvement of software and handling of any unanticipated problems. His ability to work in a multifaceted manner and to focus on detail has been a great asset to the project's performance. Further, Moustafa got involved in the design and disembarkation of the tool.

D. *Marwan Amin*

From its inception, Marwan has been working on improving the methodology defined in the Project Proposal through intense research and incorporating the latest trends and best practices into the project plan. He fully followed the roadmap, overseeing the execution and monitoring the progress by the set milestones. Furthermore, the process would also involve installing the operating system (Kali Linux) and setting up the Prometheus server. Marwan saw that the project needed improvement areas, and he made the required changes to improve the project results. In addition to his organizational role, Marwan produced the essential paperwork and files as well.

E. Mohamed Kechaou

As outlined in the original Project Proposal, Mohamed handled software tools configuration and analysis of data. He actually carried out the research, configured the required software tools, and he also tested the device. He pinpointed the ZeroTier configuration and carried out elaborate testing that revealed the truth about the system's performance, effectiveness, and stability. Moreover, he provided technical support for scripts and reports, configuring Raspberry PI, and system security; the system presentation slides were designed by him, and troubleshooting the system was his part. Also, he helped to enhance communication among the team members, making use of a collaborative and productive working environment. Mohamed's contributions were significant in steering the project towards the designated success.

VII. Discussion

This project presents a novel and advanced monitoring system for networks, that is based on microcomputers and advanced software techniques to ensure accurate and effective monitoring without compromising the size of the devices or the computational demands. The remote network monitoring system deployed on Raspberry Pi running an Apache server, along with Prometheus, Node Exporter, SNMP Exporter, Black Box Exporter, and Grafana, provides exclusive system access through ZeroTier, preventing the fast-growing need for effective monitoring solutions.

It was the evaluation of these frameworks that showed the superior performance of these methods compared to traditional methods that allow SSH and ZeroTier to run through, consequently improving the system's versatility and accessibility. On the other hand, the project realized that real-world experiments are required to study the performance under different conditions, such as users, network configurations, and device types, which may indicate the requirement of scalability in diverse network environments. Future research work will include studying system behavior in large networks and analyzing system performance under various loads and network densities.

Constant improvements are needed to adapt to new technologies and threats. The development of future features includes predictive analysis with machine learning algorithms, improved security features like HTTPS encryption within Promstack, and integration with smart devices for a unified monitoring network. Although the limitations of such a project are recognized, this project makes it possible to promote monitoring in the area of networks by offering a practical and efficient solution, which paves the way for more complex and versatile developments by researchers and practitioners who will build on these innovations.

VIII. Conclusion

This project introduced a novel technique for monitoring by constructing a device that can monitor remotely utilizing microcomputers and specialized software. In addition, it demonstrated how to create a very effective monitoring system with a small footprint and minimal processing demands. It has been discovered that these network monitoring and data visualization frameworks are highly efficient and accurate as compared to conventional methods. They drastically improve efficiency in monitoring, metric accuracy, and visualization clarity. Through the use of SSH and ZeroTier secure remote access mechanisms, the adaptability and remoteness of our system for global monitoring and administration of these systems have been significantly broadened.

Through this project, a significant contribution is made to the world of network monitoring, as the solution uses real-world data that are also practicable and efficient. The Promstack frameworks, implemented on the monitoring device based on the Raspberry Pi, bring a step forward in network monitoring technology. However, personnel challenge academic research and practical implementation has the research implications of network management that will be based on scalability in larger network environments and continuous development in response to technology and security needs.

This project is the building block for the developments to come in terms of network performance and security whose main objective is the optimization and implementation of network monitoring practices

References

[1] M. Maclean, " Raspberry Pi Computing: Monitoring with Prometheus and Grafana," Leanpub, 2017. [Online]. Available: *https://leanpub.com/rpcmonitor/read* [Accessed: Apr. 11, 2024].

[2] "Prometheus: Overview", Prometheus Documentation, 2024. [Online]. Available: *https://prometheus.io/docs/introduction/overview/#overview* [Accessed: Apr. 11, 2024].

[3] "Grafana: Latest Documentation", Grafana Documentation, 2024. [Online]. Available: *https://grafana.com/docs/grafana/latest/* [Accessed: Apr. 11, 2024].

[4] "ZeroTier Documentation", ZeroTier, Inc., [Online]. Available: *https://docs.zerotier.com/* [Accessed: Apr. 11, 2024].

[5] Raspberry Pi Foundation. "Raspberry Pi 4 Model B Specifications." Raspberry Pi. [Online]. Available: *https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/* [Accessed: Apr. 11, 2024]

[6] Zabbix. "Zabbix Manuals." [Online]. Available: *https://www.zabbix.com/manuals*. [Accessed: Apr. 11, 2024].>>>>>>>>>>>>>>>>>>>>>>>>>

[7] LibreNMS. "LibreNMS Documentation." [Online]. Available: *https://docs.librenms.org/*. [Accessed: Apr. 11, 2024].

[8] Observe. "Node Exporter Integration Documentation." [Online]. Available: *https://docs.observeinc.com/en/latest/content/integrations/node-exporter/node-exporter.html*. [Accessed: Apr. 11, 2024].

[9] A. KUMAR K., V. B S, and V. B V, "REAL TIME MONITORING OF SERVERS WITH PROMETHEUS AND GRAFANA FOR HIGH AVAILABILITY," International Research Journal of Engineering and Technology (IRJET), vol. 6, no. 4, Art. no. e-ISSN: 2395-0056, Apr. 2019, [Online]. Available: *https://www.irjet.net/archives/V6/i4/IRJET-V6I41092.pdf*

[10] "Performance Monitoring with Prometheus and Grafana Release 1.4.0," 2020. Accessed: Apr. 29, 2024. [Online]. Available: https://performance-monitoring-with-prometheus.readthedocs.io/_/downloads/en/latest/pdf/.

[11] Squadcast, "Prometheus Blackbox Exporter: Guide & Tutorial," Available: https://www.squadcast.com/blog/prometheus-blackbox-exporter [Accessed: Apr. 20, 2024].

[12] GitHub, "Monitoring Raspberry Pi using Prometheus and Grafana," Available: https://github.com/thundermagic/rpi_monitoring_with_prometheus [Accessed: Apr. 20, 2024].

[13] ZeroTier, "ZeroTier - Global Area Network," Available: https://www.zerotier.com/ [Accessed: Apr. 20, 2024].

[14] M. Kaczmarek and W. Baszun, "Monitoring Systems Comparison: Zabbix, Nagios, and Prometheus," 2018.

[15] Hackster.io, "Remote Access to Home Network Using ZeroTier," Available: https://www.hackster.io/news/remote-access-to-home-network-using-zerotier-b4d1d4d7d4d4 [Accessed: Apr. 20, 2024].

[16] Cybersecurity Insiders, "Securing Remote Access with ZeroTier," Available: https://www.cybersecurityinsiders.com/securing-remote-access-with-zerotier [Accessed: Apr. 27, 2024].