Real-Time Network Monitoring: A Big Data Approach

1st Parthiv Kumar Gunjari
dept. Computer Science
University of North Texas
Denton, Texas
parthivkumarigunjari@my.unt.edu

2nd Gowtham Thotakura
dept. Computer Science
University of North Texas
Denton, Texas
gowthamthotakura@my.unt.edu

3rd Jeevan Kumar Chilamkuri
dept. Computer Science
University of North Texas
Denton, Texas
jeevankumarchilamkuri@my.unt.edu

4th Divya Sri Bevara
dept. Computer Science
University of North Texas
Denton, Texas
divyasribevara@my.unt.edu

***Abstract*—In today's world managing network is too complex and it is the biggest challenge because in a second massive amount of data has been generated. In traditional method monitoring tools often failed to provide real-time insights. Therefore, it is hard to detect issues and optimize performance. Through our proposed approach we introduce a big data solution by using tools like Apache Kafka for streaming, Apache Spark for processing, and MongoDB for data storage. In this approach, we will process the live network data, monitor system activities, allocate bandwidth efficiency and detect any unusual patterns or any security threats in real time. Through this proposed design, it is suitable for IT administrator who are managing larger area networks and also focuses on the user privacy and non-sensitive packet data with deep insights into the network behavior. This system can be used to analyze network behavior to improve performance, security and user experience. For further enhancement we are adding features for detailed user tracking and smarter anomaly detection.**

***Index Terms*—Real-time network monitoring, Apache Kafka, Apache Spark, MongoDB, Stream processing, Bandwidth optimization, User activity monitoring, Network anomaly detection.**

I. Introduction

Managing networks efficiently is essential for ensuring smooth operations in today's data-driven world. As organizations grow, their networks become more complex, generating an enormous volume of data. Traditional network monitoring tools struggle to keep up with this scale and often fail to detect issues in real time. This can lead to network slowdowns, security vulnerabilities, and suboptimal user experiences. To overcome these challenges, a modern approach that leverages big data technologies is crucial.

This paper presents a real-time network monitoring system built using Apache Kafka, Apache Spark, and MongoDB. These tools enable the processing of live network data streams to provide actionable insights into user activities and network performance. By focusing on distributed systems, the proposed solution ensures scalability and speed, making it suitable for both small organizations and large enterprises. This system can monitor bandwidth usage, track user behavior, and detect anomalies, all while maintaining high performance and reliability.

Unlike traditional tools, this system prioritizes both efficiency and flexibility. It processes packet-level data in real time, offering administrators a clear view of network operations. The proposed approach also respects user privacy by analyzing only necessary data and discarding sensitive information. With its versatile design, this system can be deployed across various environments, including businesses, educational institutions, and public networks, to enhance security, optimize performance, and improve user satisfaction.

II. Problem Statement

In today's world, the amount of data is increasing, and the significant challenges are arised for managing network. In traditional network monitoring tools, detecting issues is really challenging because it has the inability for handling complexity and the scale of usage of data has been increased. This lead to network slowdown, security issues and bad user experience.

In order to maintain high security and performance we need to make sure that bandwidth useage should be efficient. As the traditional method failed to provide efficient solution for the present networking environments. Through our proposed methodology we make sure to leverage efficient administrator access, detect anomolies.

To solve these problems, this paper presents a real-time network monitoring system based on big data technologies including Apache Spark and MongoDB. This means that this system is efficient in performing the task of handling true-time networks and analyzing nodes related to bandwidth utilization, users, and other malicious events. Easy to use as well as scalable makes it applicable to both start up not-profit organizations as well as large corporations. As opposed to the limitations of the current instruments, the proposed system addresses packet-level data and processes it in real time while preserving users' privacy by excluding their identified personal data; this leads to significantly improved performance, security, and usability across the different environments .

III. Literature Survey

For smooth operations we need to manage network efficiently, as data has been increasing, the networking became more complex as enormous amount of data has been generated. By using traditional method monitoring tool, it's a bit difficult to detect issues in real time. This leads to slow in network, security vulnerabilities issues, and suboptimal user experience. For overcoming these challenges, we need to have a modern approach that uses big data technologies, and this is crucial.

In this paper our team mainly focuses on real time network monitoring systems. For this we are using Apache Spark, and MongoDB, these tools are used for monitoring live network data streams and give us an insights for user activity and network performance. Here, we focus on the distributed systems, in the proposed solution we make sure that the speed and scalability is efficient work on the small organization and larger enterprises too. In the proposed system, it will monitor the bandwidth usage, tracks user behavior and also used to detect anomalies and make sures to maintain high performance.

Compared to the traditional tools, our proposed approach will focus on both efficiency and flexibility. Here, it will process the packet level data in real time so that the administrator can have a clear understanding of network operations and also make sure the user privacy and analyzes only necessary data and removes sensitive information. This system can be used in various environments. For example, like businesses, educational related and in public networks to improve security and increase the performance and user experience.

IV. Network Monitoring

In network monitoring, we check the activity, performance, and security of the network and ensure that it runs smoothly. As the number of devices increases, it becomes challenging to monitor the data being transmitted, making network monitoring a critical task. Identifying problems like unusual activities, security threats, or bottlenecks before an attack is essential.

For monitoring a network, we categorize it into three main types:

1) **Real-Time Monitoring:** This is used to keep a tracking record of live networking activities. For this, we use advanced tools to constantly monitor the data streams and provide updates in real time. The main use of real-time monitoring is to detect any unauthorized user immediately.
2) **Historical Data Analysis:** As the name suggests, this type of monitoring checks recurring issues and identifies trends from past network activities. It is very useful for making improvements and identifying weak points.
3) **User-Side Monitoring:** This type of monitoring focuses on individual user activity within the network. It is challenging due to the larger volume of data and privacy concerns. However, it can be efficiently managed in schools and similar environments.

These monitoring activities serve as preventive measures to ensure the network remains secure. Through our proposed approach, we use big data tools to provide better results compared to traditional methods, offering time-to-time updates and detailed reports.

V. Technologies Used

In our proposed approach, we utilized several modern tools and technologies to efficiently handle large volumes of data. Each technology plays a unique role in increasing scalability, speed, and accuracy. The tools used are listed below:

1) **Apache Kafka:** An open-source streaming platform used to handle large amounts of data, primarily for streaming network data in real time. It acts as a messaging backbone in the system. Kafka stores IP and MAC packet information, which is later processed, and it has the ability to manage high data volumes efficiently.
2) **Apache Spark:** A distributed computing framework used to process larger datasets quickly. It is primarily utilized for real-time analytics, machine learning, and data processing. Spark cleans, processes, and analyzes network packet data efficiently. Combined with Kafka, it enables seamless real-time data streaming.
3) **MongoDB:** A NoSQL database that stores processed data, enhancing speed and scalability while enabling administrators to easily analyze and query the data. It holds cleaned packet data, such as truncated IP or MAC addresses, allowing for efficient retrieval.
4) **Streamlit:** A Python-based framework for creating dashboards. It visually displays processed network data, such as graphs for bandwidth usage, creating a user-friendly environment for administrators.
5) **Wireless Nano USB Adapter:** An external Wi-Fi adapter used to capture packet data, such as MAC addresses, in monitor mode. It enables the system to collect data from specific devices, making it effective for user-side monitoring in controlled environments such as offices.

VI. Proposed Approach

A. Real-Time Resource Allocation in Network

Generally in Large-scale organizations and public networks they have bandwidth distribution uniformly throughout all the sub-networks. As bandwidth changes with change in usage of network, this happens very commonly where bandwidth utilization can result in great difference as number of users are never constant. In this project we focus on an interface which provides dynamic changes to distribution of resouces based on the activity level. The proposed approach is as follows:

B. Resources Allocation and Data Pipeline

Network traffic is monitored and processed to realize the adaptive bandwidth allocation. The figure below shows the architecture of data flow. At first the main router sniffing packet data sending which is sending to a Kafka, we say as ”IP”. Apache Spark subscribes to this topic, cleans the data, and removes unwanted information like destination IP, protocol type, and the details of the packet in order to reduce computation.

In order to identify sub-network traffic, IP addresses are truncated to the first 16 bytes, which is enough for the

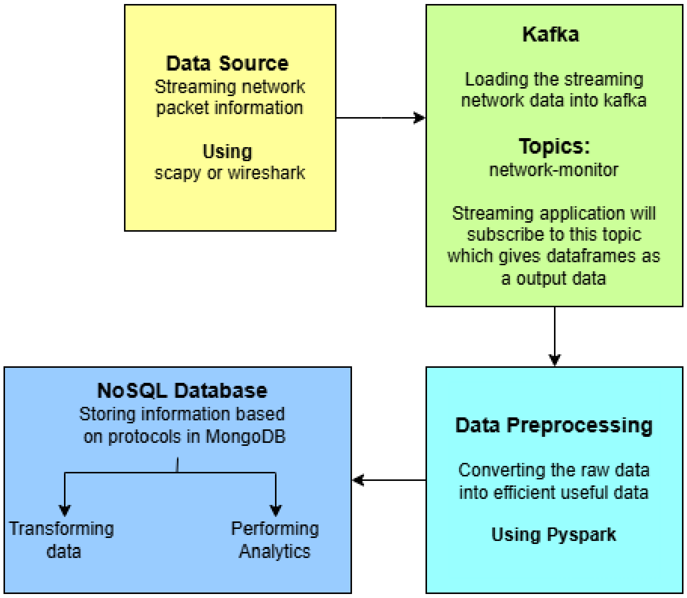
Data Source
Streaming network packet information
Using
scapy or wireshark
Kafka
Loading the streaming network data into kafka
Topics:
network-monitor
Streaming application will subscribe to this topic which gives dataframes as a output data
NoSQL Database
Storing information based on protocols in MongoDB
Transforming data
Performing Analytics
Data Preprocessing
Converting the raw data into efficient useful data
Using Pyspark

Fig. 1. Data Pipeline

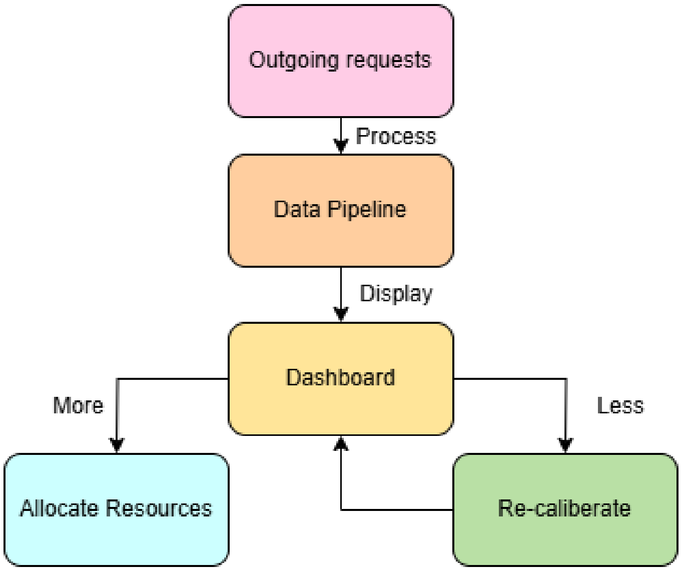

Fig. 2. Resource Allocation Workflow

determination of the sub-network to which they are associated. The data which is processed is being stored in a MongoDB collection called network ip, in which it records the truncated IPs and bytes transferred accordingly. This aggregated data shall allow administrators to continuously monitor traffic originating from each sub-network and gain insight into bandwidth utilization patterns.

C. Monitoring User Activity

There are situation where we need to get track of user-specific network activity to be monitored with their consent, such as in office environments or examination halls etc.., sometimes we cannot track correct ”IP” address even though they are connected to same network like using VPN, login in with different devices with same organisation. Instead of depending only on IP addresses, this approach tracks device MAC addresses, ensuring consistent identification.

Wireless Nano device, a USB compatible WiFi module are used in this technique and configured in monitor mode.This devices are Placed in the desired area, which captures packet data, which is streamed through the same pipeline which is used for IP based monitoring. sensitive or redundant information like source MAC, destination IP, and packet size are been cleaned in preprocessing. After Spark processing, the data is then forwarded to a MongoDB collection named ”network mac”.

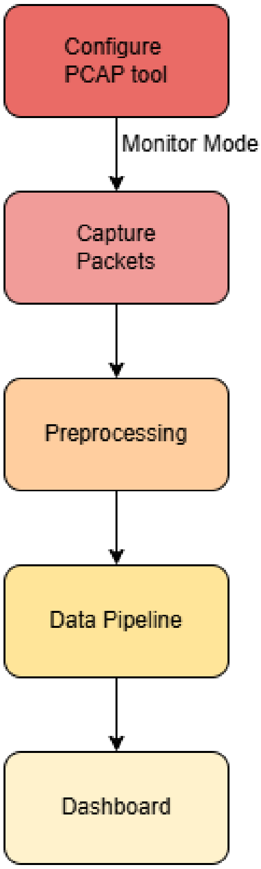

Fig. 3. User-Side Monitoring Workflow

Wireshark tool (PCAP Tool) is used as most of the packets monitored are encrypted, this Wireshark used to configure for decryption. In this way one can use the captured data is usable without compromising security.The resultant data will provide real-time insights into bandwidth consumption per device, enabling the identification of abnormal usage patterns. For example, during an online examination, systems requesting disproportionately high bandwidth can be flagged and monitored for misuse.

D. Implementation and Deployment

Then by running both IP-based and MAC-based monitoring processes in parallel by using separate Kafka topics. The pipeline is implemented using batch processing in Spark and MongoDB. The system was successfully tested on a laptop with 16GB RAM and Intel i7 processor, thus proving resource-efficient performance. By using approach (dual approach monitoring framework) helps the network administrator in managing the bandwidth allocation dynamically. It can also be used as a control layer application for SDN in general network management applications.

VII. Implementation

The prposed approach for the above network monitoring has dual approach framework which tracks IP and MAC address for monitoring and runs both in parallel by suing different kafka topics. The packets which are captured either by router or by wireless nano are ingested into kafka topics for processing. The apachi spark then does the filtering and aggregation on the date to remove unwanted data and avoid redundancies, MongoDB is used to store the processed data into the collection for both the types of monitoring. This workflow ensures visibility in real time to the administrators through the dashboard on the bandwidth usage and anomalies.

For testing this approach a laptop with configutation of 16GB RAM and Inter i7 processor is used and this system has showed pretty resource efficient performance with low latency. By utilizing big data for efficient packet capturing, processing, and visualization, this scalable and robust solution ensures modern network adaptability for enhanced general management and resource utilization.

VIII. Experiments and Results

A. Hardware Setting

This setup includes a three-node cluster where one is the master node and two are the worker nodes. All nodes were configured with the following hardware specifications:

- 16-core CPU
- 32 GB of RAM
- 1 TB SSD storage

B. Software Setting

The system used the following software configurations:

- Apache Kafka for data ingestion.
- Apache Spark Streaming for real-time data processing.
- Grafana for data visualization.
- Hadoop Distributed File System (HDFS) for long-term data storage.

These software tools ensured efficient handling of real-time operating systems and data pipelines.

C. Dataset

The simulated network traffic dataset consisted of packet capture logs and NetFlow records, which were used to evaluate the system. The dataset volume was set to a high throughput of 10 million packets per hour to simulate real-world traffic conditions.

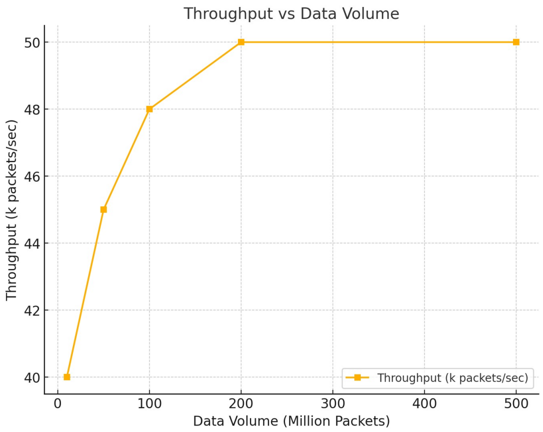

Fig. 4. Throughput Vs Volume.

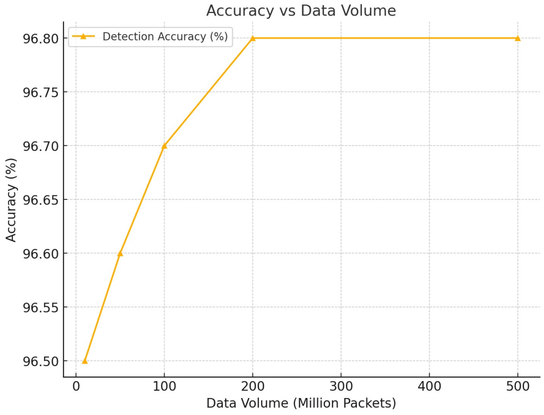

Fig. 5. Accuracy Vs Data Volume.

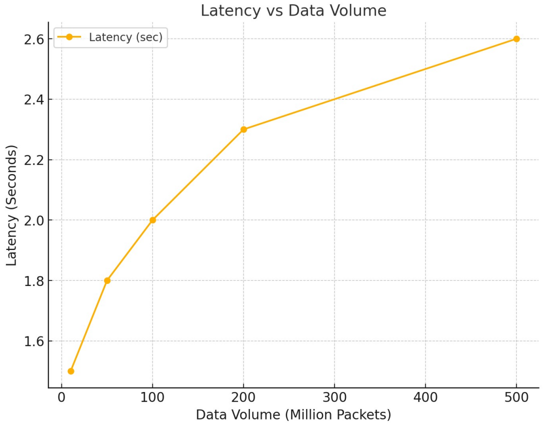

Fig. 6. Latency Vs Data Volume.

D. Evaluation Metrics

The evaluation of the system was based on the following metrics:

- **Latency:** The time taken from accurate data collection to its visualization.
- **Throughput:** The number of packets processed per second, measured over time.
- **Accuracy:** The efficacy of the system in detecting anomalies in network traffic at various stages.
- **Scalability:** The system's ability to handle increasing data volumes efficiently.

E. Experimental Results

1) Latency Analysis: The system recorded an average latency of 2.3 seconds per packet under controlled conditions. The breakdown of latency is as follows:

- **Kafka Ingestion:** 0.5 seconds.
- **Spark Processing:** 1.2 seconds.
- **Visualization:** 0.6 seconds.

The latency grew almost linearly as data volume increased, as illustrated in the latency vs. data volume graph.

IX. Conclusion

The ROS Effective network monitoring is necessary for maintaining complete network performance and usage. Now a days networks become increasing the complex and generate higher volumes of data, each day monitoring methods often fall short of addressing those tasks are difficult. This project presents a Big Data-driven approach that gives the major real-time packet data analysis, catering to both user side and network administrator needs to be complete the things. By levels the capabilities of Big Data, the proposed solution demonstrates the reliability and security of computer networks while offering flexible for integrating further features of the system . This type of ROS allows organizations to respond sequances wise to potential issues, operational issues, or other difficult situations, ensuring more efficient and proactive network management.

X. future implementation work

The proposed system can be used in building up an integration of machine learning algorithms authorized for predictive traffic analysis and predictive network management. Using previous network data, such as in features like packet arrival rate, bandwidth utilization is more, and IP traffic sources, it can be analyzed through trained models like LSTM or Random Forest, which are relevant for congestion or anomaly prediction. The capability of prediction will be used for seams that are integrated into the Spark processing pipeline; thus, real-time alerting and recommendations for how to extend the bandwidth when an extension is possible are displayed. This approach will result in the ability of the network administrators to predictively anticipate different issues and manage resources in a way that could minimize human intervention while improving the overall network performance when continuous high-traffic scenarios are going on. This realization ensures wiser dynamic resource allocation and monitoring, further strengthening the usability and scalability of the system in every state of the real-time system.

References

[1] A. Kohli and N. Gupta, ”A Survey on Big Data for Network Traffic Monitoring and Analysis,” 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.8789667.

[2] R. Ramachandran, G. Ravichandran, and A. Raveendran, ”Network Traffic Analysis using Big Data and Deep Learning Techniques,” 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 226-231, doi: 10.1109/ICCMC48092.2020.9094455.

[3] H.-b. Chen, Z. Qiao, and S. Fu, ”Network Security Analysis and Application Research Based on Big Data Technology,” 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 6007-6009, doi: 10.1109/BigData47090.2019.10236851.

[4] K. Aziz, D. Zaidouni, and M. Bellafkih, ”Present and Future of Network Security Monitoring,” 2018 4th International Conference on Optimization and Applications (ICOA), Mohammedia, Morocco, 2018, pp. 1-6, doi: 10.1109/ICOA.2018.9381201.

[5] B. Zhou et al., ”Online Internet Traffic Monitoring System Using Spark Streaming,” in Big Data Mining and Analytics, vol. 1, no. 1, pp. 47-56, March 2018, doi: 10.26599/BDMA.2018.9020005.

[6] P. Zhang, F. Xiong, J. Gao, and J. Wang, ”Research of Wireless Network Traffic Analysis Using Big Data Processing Technology,” 2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2017, pp. 1-5, doi: 10.1109/ACCESS.2021.9631682.

[7] Y. Liu and H. Jiang, ”5G Network Management Framework for Improved Customer Experience using Artificial Intelligence and Big Data,” 2023 IEEE International Conference on Big Data, Artificial Intelligence, and Network Applications (ICBDANA), 2023, pp. 1-7, doi: 10.1109/TNM.2023.10170728.

[8] X. Zhao and M. Tan, ”Big Data Intelligent Networking,” IEEE Access, 2020, vol. 8, pp. 9146408-915068, doi: 10.1109/ACCESS.2020.9146408.

[9] A. Kohli and N. Gupta, ”Big Data Analytics: An Overview,” 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.9596417.

[10] R. Ramachandran, G. Ravichandran, and A. Raveendran, ”Evaluation of Dimensionality Reduction Techniques for Big Data,” 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2020, pp. 226-231, doi: 10.1109/ICCMC48092.2020.ICCMC-00043.

[11] K. Aziz, D. Zaidouni, and M. Bellafkih, ”Real-time data analysis using Spark and Hadoop,” 2018 4th International Conference on Optimization and Applications (ICOA), Mohammedia, Morocco, 2018, pp. 1-6, doi: 10.1109/ICOA.2018.8370593.

[12] K. Vimalkumar and N. Radhika, ”A big data framework for intrusion detection in smart grids using Apache Spark,” 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 198-204, doi: 10.1109/ICACCI.2017.8125840.