Logging and Troubleshooting in Network Devices

[1]Praveen Bharade, [2]R Sindhu Rajendran

[1]Undergraduate Student, [2]Assistant Professor
Department of Electronics and Communication Engineering
Rashtreeya Vidyalaya College of Engineering

***Abstract*- Wireless network devices such as access points (AP) are crucial elements in modern networks that enable wireless device connectivity. The devices including smartphones and laptops can connect to a wired network or Internet thanks to access points, which enable wireless communication. However, access points are not impervious to issues including performance degradation, connectivity issues, and security breaches that could impair network performance and user experience. A lot of network components, including switches and routers, are present in large IP networks. These devices generate enormous amounts of logging data. Identifying network issues through analysis of this data is both a difficulty and an opportunity. Effective logging and troubleshooting techniques are essential to ensuring smooth network operation and optimum performance by immediately identifying and resolving these issues.**

***Index Terms*- logging; access points; wireless network; network management; troubleshooting.**

I. INTRODUCTION

In today's world where almost every work needs internet to complete which is provided with the large scale of IP networks. These networks contain a lot of devices which generates very large amounts of logging data [1], [2]. These logs don't need to be stored within the system permanently because the logs are generally created to keep track of the records which the user or any system which is using the internet. A relatively small number of these logs will be crucial in identifying and analyzing the problem.

Logging and troubleshooting are essential components of network device management and maintenance since performance degradation, connectivity issues, and security breaches can occur at any time. In addition to debugging and troubleshooting, logging can also be used to monitor network performance and identify potential bottlenecks or other problems that may affect performance. Network administrators need efficient logging and troubleshooting tools to correctly identify and resolve these issues, guarantee optimal network performance, and enhance user experience. Logging is the process of gathering information about network activities, events, and performance indicators that may be utilized for analysis and troubleshooting. On the other side, troubleshooting is the process of locating and resolving problems in network devices and systems. Effective troubleshooting and logging tools can help administrators in swiftly identifying and fixing issues, minimizing downtime, and enhancing network performance. As a result, implementing a logging system that can effectively store and analyze the necessary logs while discarding out the irrelevant ones is crucial. One of the most popular methods for logging in network devices is the syslog protocol, which enables devices to transmit log messages to a centralized syslog server for storage and analysis. Information regarding network events, device status, and error messages, among other things, can be found in syslog messages.

However, the restricted storage capacity of the log buffer, which can lead to the loss of crucial log data, is one issue with the deployment of syslog in network devices. Furthermore, network administrators may find it difficult to manually analyze and determine the potential problems due to the huge amount of log data that network devices generate. An effective logging and troubleshooting system is therefore required, one that can handle the large amount of log data produced by network devices and deliver precise and timely analysis of network events.

In this paper, the major focus is on logging and troubleshooting in network devices, which includes routers, switches, and firewalls. Here we propose a comprehensive framework for logging and troubleshooting that can improve network management and troubleshooting efficiency. The principal idea includes a logging module that captures various events and metrics from network devices and a troubleshooting module that analyzes the logged data to identify and resolve issues.

II. PROBLEM STATEMENT

To troubleshoot the issues on network devices, network admin has to enable debug logging to get more information about the state of the device. However, in the current deployments enabling debug for one facility enables debug-logging level for all the sub-modules and to avoid missing out of the important logs when submodules start logging at debug level, as the log buffer is rolled over upon reaching certain fixed size. This makes troubleshooting difficult as events like network outage, packet drops, flaky network or authentication failure may have occurred in the past and logs related to these events would have been rolled over.

III. PROPOSED SOLUTION

As the problem stated above which is being faced while storing and managing the logs that are generated by network devices during certain operations being carried out by them after connecting to internet, there are mainly two important solutions have been proposed in this paper which will be discussed further.

Subcategory Level Logging

Different types of logging levels present in the system, those are when arranged into increasing order based on the amount of logs they generate can be classified as emergency, warning, notice, alert, critical, error, informational and the highest level being debug. If the network admin wants to observe more logs related to WebSocket or telemetry, instead of enabling the highest level of log for the facility which has a huge number of files which are related to many other operations, by using the proposed architecture the admin can create different types of subcategories and include the files related to that subcategory and enable the required log level only for that subcategory keeping the log level of other files unchanged from that facility.

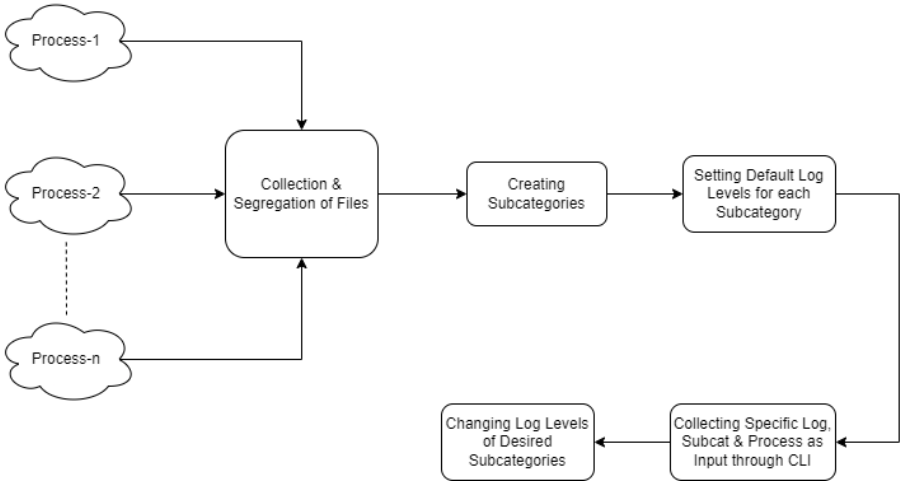

Fig. 1. Methodology of Subcategory Level Logging

Fig. 1 depicts the brief methodology of one of the proposed solutions for assigning higher level logs only for specific subcategories whose logs are needed to determine and analyze the issue instead of enabling the higher-level logs for all modules.

As many industries which manufacture the network devices including routers, switches and access points use syslog server to store the logs, there will be a log buffer which is maintained that will store the logs created by the system during any operation performed by using that network device. The main advantage of creating subcategories is that the user can get the required logs only and the other logs will be generated as usual without being affected with the additional logs of the other subcategories for which the user has set the log level.

There will be many processes running in a network device like a process for the synchronous protocols being used in that device for example Synchronous Transport Module(STM) that is a fiber optic network standard and considered as main building block for Synchronous Digital Hierarchy(SDH), another process for service access point(SAP) which is used in identifying the label for network endpoints used in Open System Interconnection(OSI) networking, a process that runs continuously in the background and perform functions required by other processes which is also called as daemon and many more. As shown in the Fig. 1, the files which belongs to the respective processes are collected and those files are segregated based on many factors such as the probability of an event being triggered more frequently, while observing logs of the files of one process, it is necessary to observe the logs of the files with similar functionality but belongs to other process. So, this is also one of the reasons for collecting all the files of different processes and segregating as per the above-mentioned factors. The segregated files are then collected under the defined subcategories based on the similar functionalities. The major advantage of doing this is when the admin wants to see the logs of particular functionality, then that can be achieved by observing the subcategories and setting the required log level for the subcategory which corresponds to the admin requirement so that the other functionality logs don't interfere the required logs which will become hectic for the admin while debugging the issues related to the network. This solves the major problem of setting the higher log level such as informational and debug for all the entire process or facility which can lead to the roll over of the logs in the log buffer due to its limited size.

The important things to be taken care while implementing this solution is to have a thorough understanding and knowledge about the files which are present under each process and their functionalities. This solution can also be improvised by again customizing the setting of log levels specific to processes within subcategories which means changing the log levels of the files which are related to specific processes under each subcategory.

Event-Based Logging

Event-based logging is a type of logging mechanism in which events or actions are logged as they occur on a network device or system. Here, the system generates a log entry each time a predefined event or activity occurs, such as a user login, network connection, or system error. These logs are typically stored in a ring buffer whose design will be discussed later and can be analyzed to identify trends, anomalies, and security breaches such as login failure, authentication error, roaming failure etc.

This solution enables network administrators to quickly identify and troubleshoot issues in real-time which reduces the time and effort required for manual analysis. It provides a comprehensive record of network activity, enabling administrators to track changes and monitor network performance. This solution can be used along with other logging mechanisms, such as periodic logging or continuous logging, to provide a comprehensive picture of network activity. It is particularly useful in environments where the volume of log data generated is high, such as large enterprise networks or cloud-based environments. This solution has an ability

to provide detailed insights into network activity. By analyzing these logs, network administrators can identify patterns and trends that may indicate performance issues, security breaches, or other potential problems. This information can be made use to proactively address the issues before they turn into critical which improves network performance and reduces the risk of security incidents.

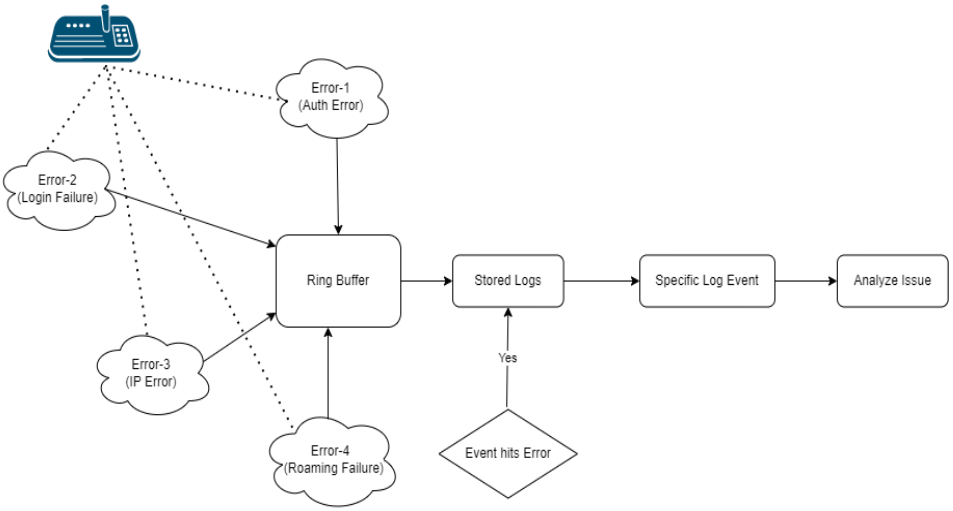

Fig. 2. Methodology of Event-Based Logging

The development of a ring buffer is the first step in achieving the objective of event-based logging. A ring buffer, also known as a circular buffer, is a fixed-sized data structure that overwrites its oldest data with the latest data when the buffer is full. It is implemented as a contiguous block of memory that wraps around at the end. The ring buffer has a fixed capacity, and it allows elements to be inserted and removed in a first-in-first-out (FIFO) order. When the buffer is full, the oldest log that is the first log will be discarded and the next log will be treated as the first log. This leads to the most recent that is new log to occupy the last position of the buffer. This process continues and so the predefined number of special logs can be stored in the ring buffer. The ring buffer should be of fixed size as it should not affect the memory management of the system and the size will be chosen according to the requirement of number of logs and use of network devices whose logs are being captured. It works by allocating a fixed-size block of memory and utilizing two pointers: one to indicate the start of the buffer, and the other to indicate the end.

Fig. 2 shows the brief methodology of how the event-based logging is achieved. At first the most important events and the most common events which occurs while operating a specific network device is noticed. This is done because as the number of times the specific event repeats, the probability of occurrence of error regarding that event increases. Then the location of operations of those events defined is discovered. Now the changes must be made while storing the generated log of that particular event. The change includes when the event is hit, the particular log will be generated and stored in the log file which is common to either the facility or the process to which it belongs. That log must be simultaneously stored into the ring buffer. The advantage of this is, as the facility or the process level log files will be filled very quickly because they store all the logs which are generated by the respective facility or process, the logs of the events occurring repeatedly or the logs of important events will be lost quickly due to the roll over of the buffer which is storing those log in the respective files. Now if these important logs are stored in a special buffer and the ring buffer will not get filled as quick as the common log file because it will store only specific logs. When any specified event hits error then the network admin can easily have the logs related to that error stored inside the ring buffer which can be very helpful while analyzing the issues.

IV. CONCLUSION

Both subcategory level logging and event-based logging are valuable tools for network administrators in managing and securing network infrastructure. Subcategory level logging provides a comprehensive record of network activity, enabling administrators to troubleshoot issues, monitor network performance, and ensure compliance with regulatory requirements. Event-based logging, on the other hand, enables real-time monitoring and provides the detailed insights into specific events and activities that are relevant to network administrators. By using a combination of subcategory level logging and event-based logging, network administrators can capture a comprehensive record of network activity while still efficiently monitoring specific events of interest. This approach can help to reduce the volume of log data generated while still providing critical information for network management and security. Overall, effective logging is a critical component of network management and security, providing network administrators with the tools they need to maintain optimal network performance and protect against security threats.

V. FUTURE SCOPE

The use of subcategory level logging and event-based logging in any network devices may become even more critical in future as wireless networks continue to evolve. With the deployment of technologies such as 5G and Wi-Fi 6, network devices will need to capture and analyze a greater volume of data in real-time. To meet this challenge, future logging mechanisms may incorporate more advanced analytics to quickly identify and respond to issues. Additionally, logging mechanisms may need to be designed to work with new wireless technologies to ensure that the unique characteristics of these networks are captured and analyzed effectively.

VI. ACKNOWLEDGMENT

The authors express their gratitude to Mr. Shreekanth Hiremath and Mr. Mahesh Dantkale for supporting this work.

REFERENCES:

1. M. Macit, E. Delibas, B. Karanlık, A. I. Sekom and T. Aytekin, "Real Time Distributed Analysis of MPLS Network Logs for Anomaly Detection", in NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, April 25-29, 2016,J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
2. K. Yamanishi and Y. Maruyama, "Dynamic syslog mining for network failure monitoring," in Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005, R. Grossman, R. J. Bayardo, and K. P. Bennett, Eds. ACM, 2005, pp. 499–508.
3. T. Qiu, Z. Ge, D. Pei, J. Wang, and J. J. Xu, "What happened in my network: mining network events from router syslogs" in Proceedings of the 10th ACM SIGCOMM Internet Measurement Conference, IMC 2010, Melbourne, Australia - November 1-3, 2010, M. Allman, Ed. ACM, 2010, pp. 472–484.
4. Risto Vaarandi, Bernhards Blumbergs and Markus Kont. "An unsupervised framework for detecting anomalous messages from syslog log files" published in NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, 23-27 April 2018.
5. C Roja and P N Jayanthi, "Syslog Daemon for Security Event Monitoring using UDP Protocol" published in 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 12-14 June 2019.
6. Amit Aeri and Shyam Tukadiya, "A comparative study of network based system log management tools", published in 2015 International Conference on Computer Communication and Informatics (ICCCI), 08-10 January 2015.
7. Jian-hua Huang, Man-qi Zhang and Yuan-long Jiang, "The design and implement of the centralized log gathering and analysis system" Published in IEEE International Conference on Computer Science and Automation Engineering (CSAE), 25-27 May 2012.
8. Karel Slavicek, Jaroslav Ledvinka, Michal Javornik and Otto Dostal, "Mathematical Processing of Syslog Messages from Routers and Switches", published in 2008 4th International Conference on Information and Automation for Sustainability, 12-14 December 2008.
9. Amit Aeri and Shyam Tukadiya, "A comparative study of network based system log management tools", published in International Conference on Computer Communication and Informatics (ICCCI) in 24 August 2015.
10. Kees M. van Hee, Zheng Liu and Natalia Sidorova, "Is my event log complete? — A probabilistic approach to process mining", published in 5th Internationa Conference on Research Challenges in Information Science, in 19-21 May , 2011.