Project Report

On

NETWORK MONITORING SYSTEM

Submitted in partial fulfillment of the requirements for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

(Artificial Intelligence & Machine Learning)

by

Ms. G NIKHITA (22WH1A6601)

Ms. G REVATHI (22WH1A6606)

Ms. R GEETIKA SRI (22WH1A6654)

Ms. B SRI VAISHNAVI (22WH1A6662)

Under the esteemed guidance of

Ms. P Anusha

Assistant Professor, CSE(AI&ML)

Department of Computer Science & Engineering

(Artificial Intelligence & Machine Learning)

BVRIT HYDERABAD COLLEGE OF ENGINEERING FOR WOMEN

(AUTONOMOUS)

(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with A Grade

Bachupally, Hyderabad – 500090

2024-25

Abstract

The Network Monitoring System is a real-time application designed to oversee and manage the performance and security of a network. It continuously tracks key parameters such as network traffic, bandwidth usage, device status, and potential security threats. The system collects data from various network components, providing administrators with a comprehensive view of network health and performance. It includes tools for monitoring network traffic, detecting anomalies, and generating alerts for potential issues. The system ensures prompt identification and resolution of problems, improving network efficiency and security. Additionally, the system can be extended to offer detailed reports, historical data analysis, and support for remote management, making it ideal for organizations aiming to maintain robust and secure network infrastructures.

Problem Statement

Develop a state-of-the-art **Network Monitoring System** that provides a secure, scalable, and user-friendly platform to revolutionize network performance management. This system will offer **real-time traffic analysis, advanced anomaly detection, detailed device health monitoring, and proactive alerting mechanisms**, enabling organizations to anticipate and address issues before they escalate. Designed for seamless integration with existing infrastructure, it ensures effortless scalability to handle large, distributed, and complex networks while maintaining top-notch security standards. With features like automated backups, swift recovery options, and intuitive dashboards, the system empowers users to optimize network performance, reduce downtime, safeguard against evolving threats, and create a robust, efficient, and highly reliable network environment that adapts to future demands.

.

Functional Requirements

1. **Real-Time Monitoring:**
    - The system must monitor network traffic, bandwidth usage, and device status in real-time.
    - It should be able to track key metrics such as latency, packet loss, and network throughput.
2. **Traffic Analysis:**
    - The system should analyze and display detailed reports on network traffic, including inbound and outbound data.
    - It should provide insights into traffic patterns, peak usage times, and potential bottlenecks.
3. **Alerting Mechanism:**
    - The system must generate alerts for network issues such as high latency, packet loss, unauthorized access attempts, or device failures.
    - Alerts should be configurable based on thresholds set by the network administrator.
4. **Device and Node Monitoring:**
    - The system should be able to monitor the status and performance of all devices and nodes within the network.
    - It should support SNMP (Simple Network Management Protocol) for device communication and monitoring.
5. **Anomaly Detection:**
    - The system must detect network anomalies, such as unusual traffic spikes, unauthorized access, or deviations from normal network behavior.
    - It should include machine learning or heuristic-based algorithms to identify potential security threats or performance issues.
6. **Reporting and Logging:**
    - The system should generate detailed logs of all network activities, including traffic statistics, device status, and alerts.

- It should provide customizable reporting features, allowing network administrators to generate specific reports based on various metrics.

7. **User Access Control:**
    - The system should support role-based access control (RBAC), allowing network administrators to define access rights for different users.
    - Different users should have different levels of access (e.g., administrator, viewer, operator).
8. **Dashboard:**
    - The system should provide a user-friendly dashboard displaying an overview of network health, including key performance indicators (KPIs) and real-time status updates.
    - The dashboard should allow drill-downs for detailed views of specific devices or traffic segments.
9. **Network Configuration Management:**
    - The system should allow network administrators to manage network configurations remotely.
    - It should provide a central interface to push configuration changes to network devices.
10. **Data Backup and Recovery:**
    - The system should have the ability to back up monitoring data and configuration settings.
    - It should provide seamless recovery options in case of data loss or system failure.

Non-Functional Requirements

1. **Scalability:**
    - The system should be scalable to monitor large and growing network infrastructures.
    - It should be able to handle multiple devices, traffic sources, and users without performance degradation.
2. **Performance:**
    - The system should be able to process and display real-time data with minimal latency.
    - It should not introduce significant overhead on the network it monitors.
3. **Availability:**
    - The system must be available 24/7, ensuring continuous monitoring of the network.
    - It should include failover mechanisms to ensure high availability in case of server or component failure.
4. **Security:**
    - The system should use encryption for all data transmission, including monitoring data and alert messages.
    - It should include strong authentication and authorization mechanisms to protect against unauthorized access.
5. **Usability:**
    - The system should have an intuitive user interface that allows network administrators to configure and monitor the network easily.
    - The dashboard should be customizable, providing both high-level overviews and detailed information on demand.
6. **Interoperability:**
    - The system should support integration with a wide range of network devices, operating systems, and third-party applications.

- It should support standard protocols such as SNMP, NetFlow, and syslog.

7. **Data Integrity:**
   - The system must ensure that the collected network data is accurate and reliable.
   - It should perform periodic integrity checks and maintain consistency in data storage.
8. **Maintainability:**
   - The system should be designed to facilitate easy updates, bug fixes, and enhancements.
   - It should have clear documentation for system maintenance and troubleshooting.
9. **Extensibility:**
   - The system should be easily extensible to support future features and integrations, such as additional protocols, security measures, or cloud-based services.
10. **Cost-Effectiveness:**
    - The system should be optimized for cost, balancing performance, scalability, and resource usage without incurring excessive infrastructure or operational costs.
11. **Compliance:**
    - The system should comply with industry standards and regulations regarding data protection, network security, and monitoring practices (e.g., GDPR, HIPAA, PCI-DSS, etc.).

Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_HOSTS 5
#define BUFFER_SIZE 256
void monitor_host(const char* host) {
    char command[BUFFER_SIZE];
    snprintf(command, sizeof(command), "ping -c 1 %s > /dev/null 2>&1", host);
    int status = system(command);

    if (status == 0) {
        printf("Host %s is reachable.\n", host);
    } else {
        printf("Host %s is not reachable.\n", host);
    }
}

int main() {
    int num_hosts;
    char hosts[MAX_HOSTS][BUFFER_SIZE];

    printf("Network Monitoring System\n");
    printf("=========================\n");
    printf("Enter the number of hosts to monitor (max %d): ", MAX_HOSTS);
```

```
    scanf("%d", &num_hosts);
    if (num_hosts < 1 || num_hosts > MAX_HOSTS) {
        printf("Invalid number of hosts. Please enter a value between 1 and %d.\n",
MAX_HOSTS);
        return 1;
    }
    for (int i = 0; i < num_hosts; i++) {
        printf("Enter host %d: ", i + 1);
        scanf("%s", hosts[i]);
    }
    printf("\nMonitoring Hosts...\n");
    printf("====================\n");
    for (int i = 0; i < num_hosts; i++) {
        monitor_host(hosts[i]);
    }

    return 0;
}
```

Compilation and Execution

Compile :

gcc -o network_monitor network_monitor.c

Run the executable code:

./network_monitor

Output

Reachable hosts

```
Network Monitoring System
=========================
Enter the number of hosts to monitor (max 5): 3
Enter host 1: google.com
Enter host 2: github.com
Enter host 3: localhost

Monitoring Hosts...
===================
Host google.com is not reachable.
Host github.com is not reachable.
Host localhost is not reachable.
```

Not reachable hosts

```
Network Monitoring System
=========================
Enter the number of hosts to monitor (max 5): 3
Enter host 1: google.com
Enter host 2: unknown-host.xyz
Enter host 3: localhost

Monitoring Hosts...
===================
Host google.com is not reachable.
Host unknown-host.xyz is not reachable.
Host localhost is not reachable.
```