**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

Jnana Sangam, Belagavi-590018



### Project Report

on

# ENHANCING NETWORKING MONITORING SYSTEMS BY OVERLAYING PROTOCOLS

by

**RAJASHREE**

**1VI17CS114**

**PAVITHRA K**

**1VI17CS071**

UNDER THE GUIDANCE OF

### Mr. NOOR PASHA

Asst. Professor Department of CSE, Vemana Institute of Technology



### DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,

### VEMANA INSTITUTE OF TECHNOLOGY

3RD BLOCK, KORAMANGALA, BENGALURU – 560034

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| 1. **Introduction** | **1** |
| 1. **Literature Survey** | **2** |
| 1. **Objective** | **3** |
| 1. **Design Methodology** | **4** |
| 1. **Software and Hardware Requirements** | **5** |
| 1. **Applications** | **6** |
| 1. **Outcome** | **7** |

**CHAPTER 1**

**INTRODUCTION**

A computer network is a group of network nodes that use a set of common communication protocols over digital interconnections for the purpose of sharing resources located on or provided by the network nodes. A network node is either a redistribution point or a communication endpoint such as modem, hub, switch, bridge, routers, network file systems, personal computers and so on. Network monitoring system is a scripted tool that administers a computer network for slow or failing components and that notifies the configured users in case of outages, latencies, upgrades, and other events.

With the advent of internet, network monitors can now perform variating services such as fault analysis, performance management, provisioning of networks and maintain quality of service. Internet technologies have become very important in every person’s day to day life and proactive monitoring of interconnected devices has become vital in the internet service provision business. The onset of ecommerce has reduced the prices of IoT to great extent and a typical home in suburban consists of Layer 3 router, a Wi-Fi access point, internet connected TV and remotely controlled light bulbs. As a result, security and monitoring have become a critical concern for every person with internet connectivity.

The latest generation of network monitoring systems are designed to support very specific applications. Most of them in general, are devised to monitor enterprise environments that generally consist of 100s of similar network nodes such as a network rack, routers, switches, and personal computers. Further, they are designed to support very specific applications such as identifying the device’s connectivity or latency or out-of-band analysis and so on. To obtain results they are homogenous in terms of protocol usage. For example, some systems use ICMP alone to obtain performance analysis of a huge network. Also, due to the lack of a buffer Dataset, they are not generally real-time and are quite torpid. A typical topology that consists of less than 50 network nodes such as a startup environment or a home office environment cannot afford such a stack of tools and maintaining a stable and high performance network is an up-hill task for them. As a result, most of the emerging organizations employ a separate team that act as network administrators.

**CHAPTER 2**

**LITERATURE REVIEW**

**[1]** **A transparent virtual machine monitor level packet compression network service – by Ali Hamidi, Hadi Salimi and Mohsen Sharifi. [2010]:**

This paper explains about packet compression, a ubiquitous technique for improving the performance of low speed networks such as WANs, especially effective in networks with high cost per transmitted byte such as wireless networks. It also shows how packer compression service can be used in virtualization technology in which network packet payloads are compressed and hence application communication overhead involved in virtual machines is reduced. The feasibility of development and dynamic configuration of such service and the extensive improvement of network performance over congested links is proven in the results and provided as a proof of concept.

**[2] A Monitor Tool for a Network Based on the Cambridge Ring – by Synnove Vassiliades, Michael D. Sayers and Jean M. Bacon. [1986]:**

This paper shows how networks and their demands have evolved over time and different options to monitor a network. Also, it illustrates how to validate analytic and simulate models for network topologies, detect implementation errors, performance bugs and so on. It also describes a measurement facility which has been developed to discuss the issues and practical constraints involved in its design to suggest enhancements in the hatfield network.

**[3] Design and implementation of a Web-based Internet Performance Management System Using SNMP MIB-II – by Seong Jin Ahn, Seung Keun yoo and Jin Wook Chung [1999]:**

This paper analyses various attributes required for monitoring performance of a network and how to extract them using the SNMP MIB-II and MATP protocol. It uses Java to form a web-based tool to show the current performance of a network. Although it was first of its kind, this model lacks a buffer databased between the graphical user interface and back end implementation which makes the data loading slow, especially in virtual private networks.

**[4] Network Management in the work of standards: The Role of the SNMP Protocol in Managing Networks – by Dr Katherine Jones [1991]:**

This white paper gives a high level view of the SNMP protocol, current standards in network management, MIBs, responsibilities of a SNMP manager, examples of SNMP Implementations, issues with current GUIs in SNMP, selection criteria for purchasing a network manager and various vendors. This paper was particularly useful in articulating the current proposed system.

**[5] On evaluating the differences of TCP and ICMP in network measurement – by Li Wenwei, Zhange Dafang, Yang Jinmin and Xie Gaogang [2007]:**

This paper the parameters and results of measuring host connectivity, RTT andpacket loss rate are compared between TCP and ICMP. While the accuracy of the results are higher in TCP, the time taken was significantly less in ICMP especially in the case of calculating RTT.

**CHAPTER 3**

**OBJECTIVE**

1. To instantly detect device outages or performance reductions and immediately trigger notifications via E-Mail/SMS/GUI alerts to configured users.
2. To perform polling of device statistics in real-time to measure various performance attributes such as reachability, availability, bandwidth, and latency.
3. To provide easy to view and comprehensive web-based application user interface that automatically obtains device info, uptime, real-time statistics, traffic statistics and so on.
4. To provide periodic historical reports of devices on their performance.
5. To facilitate accessing different network nodes via SSH/Telnet/Web access from the monitor itself.
6. To facilitate custom configuration of notifications to different types of users in various situations through various mediums.
7. To add support monitoring of various network nodes such as Servers, Routers, Switches, Power Distribution Units (Network PDUs), Virtual Machines, IoT devices, Cloud instances, Data Stores, Wireless Access Points, Endpoint PCs and so on.

**CHAPTER 4**

**DESIGN METHODOLOGY**

**4.1 Components:**

**4.1.1 Graphical User Interface:**

* This facilitates the user to login into the system, initiate the utility, whereby detail of the entire network is present in a table.
* This table is automatically filled with details of the network node such as MAC Address, OS and so on using LLDP protocol
* The Traffic Stats will show up eventually which are obtained from the device using ICMP protocol
* The user can also see all the performance attributes such as reachability, latency and so on present on any network node by switching between tabs.
* The user can take SSH/Web/Telnet access to the network node by clicking on Access button in the table.
* The user can update, modify, change, or configure the network nodes. This triggers a background process of history keeping. It also provides the ability to caution any change of an attribute which may not be compatible with the resource at hand.
* The GUI also shows latest alerts on the top with various color codes assigned based on the criticality of the event.
* Types of Alerts and their frequency can be configured under the Profile option.
* Coding Languages used for this Component: HTML, CSS, Bootstrap Templates, JavaScript

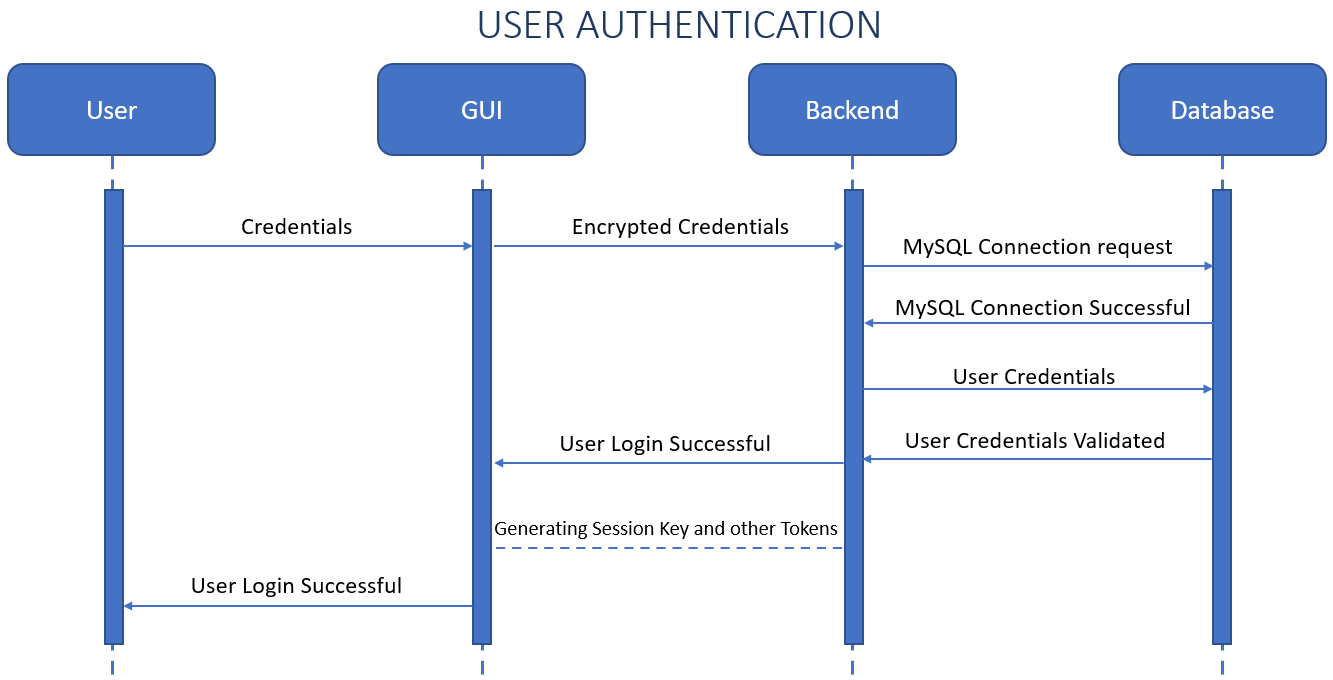
**4.1.2 Back-end Processing:**

* This module handles all functionality from finding the network nodes, attributes, querying with the network nodes and updating the database with information obtained through back end processing
* It also maintains a dynamically updated database which contains all the attributes present on every resource along with compatibility problems. As a result, it can take decisions on issues occurred and report errors accordingly.
* Various protocols are used to obtain different types of data for example LLDP for device information, SNMP for performance statistics, SSH/Telnet/HTTP for device access, ICMP for traffic statistics and so on.
* Raw IP sockets are formed and sent in various ways to determine the operating system the network nodes are using, packet filters in use and several other attributes.
* Coding Languages used for this Component: Python, Django/Flask Framework, JavaScript, pymysql framework

**4.1.2 Database:**

* This module handles acts a buffer between the backend functionality and the graphical user interface. It provides synchronization between the activities of the user and the backend
* To avoid top down triggering and querying of the network node data every time the web page is loaded or meta-refreshed, the backend functionality runs infinitely to fill various attributes of the network node by querying it using various network protocols at regular intervals.
* Simultaneously, GUI fetches data from database which results in real-time transmission at any given second.
* MySQL Tables used:
  + UserAuthentication: Deals with user login, logout and session keys
  + Devices: Deals with devices and their attributes at real time
* Languages used for this Component: MySQL Database, MySQL query Language, MySQL Workbench / MySQL CLI for diagnostics

**4.2 Sequence Diagrams:**

**4.2.1: User Authentication: **

**Fig 4.2.1 User Authentication**

**CHAPTER 5**

**REQUIREMENT SPECIFICATIONS**

**5.1 Technical Requirements:**

**5.1.1 Hardware Requirements:**

* Processor: Intel® Core i5 ™ CPU and above
* RAM: 8 GB or higher
* Hard Disk: 100 GB or higher

**5.1.2 Software Requirements:**

* Operating System: Windows 10/Ubuntu 20.04 LTS
* Architecture: 64-bit OS
* Python 3.8 or higher
* PIP Packages: RegEx, Flask, Django, Pymysql
* Database: MySQL5.7 or higher
* JavaScript 1.8.5 or higher
* Front End: HTML5, CSS3, Bootstrap4
* Code Editor: Microsoft Visual Studio Code

**Chapter 6**

**APPLICATIONS**

* Networks serve as the backbone for any enterprise. Any network outage during working hours is huge loss for the organizations.
* As a result, they employ a separate team to look after their labs by constantly logging into several system interfaces and checking their statuses.
* It is a tedious task to login to each of these nodes, check if they are reachable and check their health status constantly.
* As the enterprise grows, the numbers increases exponentially.
* Generating network performance reports
* Deploying new technology and software upgrade successfully
* Monitoring the flow of traffic with netflow
* Track user network activity