Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technologies

Diploma Thesis

Performance Monitoring of highly Complex Network Systems

Handoro Semayat Fikre

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

B.Sc. Semayat Fikre Handoro

Systems Engineering and Informatics
Informatics

Thesis title

Performance Monitoring of highly complex Network Systems

Objectives of thesis

The main objective of this thesis will be to design a performance Monitoring tool/ approach for highly complex Network Systems that can resolve some of the problems and limitations that arise by using the protocol-based approach.

Side objectives include studying and evaluating existing network monitoring tools with limitations, selecting and measuring appropriate Key Performance Indicators for the network performance monitoring architecture.

Methodology

After the literature review including the analysis of the currently used methods and techniques, the first procedure will be to do a proper evaluation of the already existing tools to present a visible gap and list out the limitations mentioned related to the protocol based approach with examples. In accordance with the limitations found, picking up KPI’S for the approach that will be presented. Learning the network management by using the simple network management protocol and integrating the idea with receiving network data which author desire to monitor and storing it, which further will be converted to javascript object notation for the monitoring system. Based off of that, building up an architecture that can resolve some of the limitations motioned with monitoring examples.

The proposed extent of the thesis

60

Keywords

SNMP , Real time performance ,ICMP-PING, Network Monitoring

Recommended information sources

Andrew Tanenbaum. 2002. Computer Networks (4th ed.). Prentice Hall Professional Technical Reference.

M. Allman, S. Floyd, and C. Partridge. RFC 2414: Increasing TCP's Initial Window, September 1998. Status: INFORMATIONAL. (p130)

William Stallings "SNMP and SNMPv2: The Infrastructure for Network Management," IEEE Communications Magazine, Vol. 36, No.3, March 1998 pp

Expected date of thesis defence

2020/21 SS – FEM

The Diploma Thesis Supervisor

Ing. Tomáš Vokoun

Supervising department

Department of Information Technologies

Electronic approval: 20. 7. 2020	Electronic approval: 21. 10. 2020
Ing. Jiří Vaněk, Ph.D.	**Ing. Martin Pelikán, Ph.D.**
Head of department	Dean

Prague on 16. 02. 2021

Declaration

I declare that I have worked on my diploma thesis titled " Performance Monitoring of highly Complex Network Systems" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on March 21, 2021 ___Handoro Semayat Fikre ___________________________

Acknowledgement

I would like to thank my supervisor Ing. Tomáš Vokoun for the patient guidance and encouragement throughout my time as his student. My appreciation also goes out to my family and friends for their encouragement and support all through my studies, completing this thesis work would have been all more difficult were it not for your support.

Performance Monitoring of highly Complex Network Systems

Abstract

Complexity of network systems is increasing and consequently the need to monitor network infrastructure has become crucial as it allows organizations to foresee potential outages and address network issues in a proactive manner to lower down any business impact. Monitoring approaches that are efficient and well suited to monitor a quite simpler network are not as efficient when used in a complex mission critical network. This thesis proposes a design of a monitoring tool (approach) that could be suitable to monitor large scale networks and resolves some of the limitations that may arise by using the protocol-based approach. It Conducted and presented an evaluation made on using a protocol-based approach (SNMPv3) to monitor large scale networks. The evaluation was conducted on a real production network environment from the perspectives of examining the protocol if it intensively uses the resource it monitors and its performance in reporting network events and issues. After observing results of the conducted evaluation and taking related factors like telemetry options that can provide different methods of communication with monitored devices and flexible interfaces where encoding of the monitoring data is performed in a text format using JavaScript object notation Into account, it proposes an architectural design with detailed working logic. The design of the system was carried out using the unified modelling language UML design diagrams. Major processes that are associated with the system to carry and transfer data were shown by using three levels of logical data flow diagrams.

Keywords: SNMP, Network Monitoring, Data telemetry, Unified modelling language, Netconf, Plug-in Architecture, RestConf, Real time performance, ICMP-PING

sledování výkonu vysoce složitých síťových systémů

Abstrakt

Složitost síťových systémů se zvyšuje a v důsledku toho se potřeba monitorovat síťovou infrastrukturu stala zásadní, protože umožňuje organizacím předvídat potenciální výpadky a proaktivně řešit problémy se sítí, aby se snížil dopad na podnikání.

Monitorovací přístupy, které jsou efektivní a vhodné pro monitorování poměrně jednodušší sítě, nejsou tak efektivní, když se používají v komplexní síti kritické pro mise. Tato práce navrhuje návrh monitorovacího nástroje (přístupu), který by mohl být vhodný pro monitorování rozsáhlých sítí, a řeší některá omezení, která mohou nastat při použití přístupu založeného na protokolu. Provedla a představila hodnocení provedené pomocí přístupu založeného na protokolu (SNMPv3) ke sledování sítí ve velkém. Hodnocení bylo provedeno na reálném prostředí produkční sítě z hlediska zkoumání protokolu, pokud intenzivně využívá prostředek, který monitoruje, a jeho výkon při hlášení událostí a problémů v síti. Po pozorování výsledků provedeného vyhodnocení a převzetí souvisejících faktorů, jako jsou možnosti telemetrie, které mohou poskytnout různé metody komunikace s monitorovanými zařízeními a flexibilní rozhraní, kde se kódování monitorovacích dat provádí v textovém formátu pomocí notace objektu JavaScript

Z tohoto důvodu navrhuje architektonický návrh s podrobnou pracovní logikou. Návrh systému byl proveden pomocí návrhových diagramů UML jednotného modelovacího jazyka. Hlavní procesy, které jsou spojeny se systémem pro přenos a přenos dat, byly ukázány pomocí tří úrovní logických diagramů toku dat.

Klíčová slova: SNMP, monitorování sítě, datová telemetrie, unifikovaný modelovací jazyk, Netconf, architektura plug-inů, RestConf, výkon v reálném čase, ICMP-PING.

Contents

List of Figures

List of Tables

1 Introduction

In the earlier days of networking, computers and the internet were used by a very limited number of people, mainly as a tool for scientific research. During that period, network management was practically an unknown term. While in today's world it's one of the widely known terms in the IT industry. This is due to the development in technology and its demand. A network monitoring system is a process by which different network nodes or components like a router, switch, PSU, Firewalls, VM's, servers and more are monitored in a continuous manner to detect performance issues, optimize their availability, and manage configurations. It is crucial as it generates reports of how the managed network is performing over a defined period. Taking account of these reports, network administrators can foresee when the organization may need to consider upgrading, implementing, or optimizing new IT services or infrastructure. In the earlier days of networking and the internet, Individuals and organizations have tried to present and introduce various ways to monitor Network infrastructure and devices. It was not until 1988 that Simple Network Management protocol or SNMP became a standard based on the requirements of the Internet Engineering Task force. In the years and decades since its initial development, SNMP has continued to be developed and refined, to the point where the protocol is now anything but "simple". In Today's world almost all network devices and components have Built-in SNMP capability and their software's have SNMP features list.

A network monitoring system generally includes software and hardware tools, these tools help by tracking different characteristics of the network infrastructure, its services and operations like its bandwidth utilization, traffic, its uptime, the temperature, and fan status of components of different nodes and more. These systems have a way of sending out notifications regarding the status of the managed nodes and services to administrators. One of the major features of a good monitoring system its ability to be more proactive than reactive which greatly helps engineers and users to identify issues at their early stages. Having a well-organized, methodical, and efficient monitoring system is pivotal as it can avert network downtime or any related failures. Since mid-2015/2016 Multiple bloggers and vendors are trying to sell the idea of replacing the legendary SNMP with other alternative protocols, although it is difficult to find a complete alternative to SNMP mainly because it has been implemented into millions of systems and software over the last 30 years. There are some tools getting into the market reducing the dependency on SNMP. Even though a few of these tools can solve some of the problems of a complex large-scale network, there are also other two starts contributing to it.

Software-defined Networking (SDN) and API technology. Today, both the IETF and the Open Config community are focusing on telemetry solutions and working on developing the best standards to run with different Data models. From what is being discussed in the industry, different vendors, IT companies and academicians are recently trying to push the notion that SNMP is dead and has no future, However the reality is that SNMP Is still here and widely used. There are a few articles, blogs, and journals written on the performance of SNMP. They tend to look like they are impartial in their conclusions and comparisons. Unfortunately, almost all their evaluations were performed in a lab environment and they often aimed at proposing an overly expensive new emerging tool or approach which from a performance point of view, is supposed to outperform SNMP. However, the evaluation conducted in this thesis is on real production network where live traffic was running to make sure that evaluation criteria and performance indicators are well tested to then propose a design (Approach).

2 Objectives and Methodology

2.1 Objectives

The main objective of this thesis will be to design a performance Monitoring tool/ approach for highly complex Network Systems that can resolve some of the problems and limitations that arise by using the protocol-based approach.

Side objectives include studying and evaluating existing network monitoring tools with limitations, selecting, and measuring appropriate Key Performance Indicators for the network performance monitoring architecture.

2.2 Methodology

After the literature review including the analysis of the currently used methods and techniques, the first procedure will be to do a proper evaluation of the already existing tools to present a visible gap and list out the limitations mentioned related to the protocol-based approach with examples. In accordance with the limitations found, picking up KPI’S for the approach that will be presented. Learning the network management by using the simple network management protocol and integrating the idea with receiving network data which author desire to monitor and storing it, which further will be converted to java script object notation for the monitoring system. Based off of that, building up an architecture that can resolve some of the limitations motioned with monitoring examples.

3 Literature Review

This chapter describes some historically significant contributions and current work particularly related to the concepts described in this thesis. The below key points and concepts will aid the reader in the future chapters of this thesis.

3.1 Network Monitoring and its value

At the present time Network monitoring is broadly spread throughout the information and technology services and industry. Being able to fine performance issues proactively has a great impact and help on determining issues depend at their initial stage. Regardless of fast performance improvements in the field of network technologies and their pervasiveness, today's computer-demanding and service-oriented applications require efficient management of networks(Kulkarni, Liu, Ramakrishnan, Arumaithurai, Wood, Fu 2018).The concept behind Network monitoring is a very extensive and meticulous topic. To thoroughly explain the concepts, this section mentions some background information.

The term network monitoring describes a range of techniques by which it is sought to observe and quantify exactly what is happening in the network, both on the microcosmic and macrocosmic time scales.(Landfeldt, Sookavatana, Seneviratne 2000) Data gathered using these techniques provides an essential in put towards:

a) Performance tuning: - identifying and reducing bottlenecks, balancing resource use, improving QOS and optimizing global performance.
b) Troubleshooting: - identifying, diagnosing, and rectifying faults.
c) Planning: - predicting the scale and nature of necessary additional resources.
d) Development and design of new technologies: Understanding of current operations and Trends motivate and direct the development of new technologies.
e) Characterization of activity to provide data for modelling and simulation in design and Research.
f) Identification and correction of pathological behavior.

3.2 Passive and Active Network Monitoring

Network monitoring can be either active or passive. In the later, the monitoring system reads data without affecting the traffic. However, active network monitoring gives the choice to alter the data on the line. (Svoboda, Ghafir, Prenosil, others 2015).

In a Passive network monitoring, what happens is capturing network traffic that flow through a network and analyzing it afterwards. Through a collection method like log management or network taps, passive monitoring compiles historic network traffic to paint a bigger picture of a network performance. The primary use for passive network monitoring is for discovering and predicting performance issues that happen at specific instances and areas of a network.

With regards to resource utilization, passive monitoring uses less resources than active monitoring which on the other hand is resource intensive. It is a useful method for analyzing a network performance event after it occurs. The good side or strength of using passive monitoring is that there is no intrusion made in to the monitored traffic and further more while performing on-line monitoring using probs attached directly to a network link the whole data with respect to the network traffic‘s is potentially obtainable (Hall 2003).

The weakness, particularly as and the volume of traffic carried and the network bandwidths increase, is that it becomes somehow hard or difficult to put up with the traffic passing through (in the processing power required both to collect the data and to carry out any contemporaneous processing) and that the volume of data collected becomes unmanageable(Jackson, Sterbenz, Condell, Hain 2002).

When it comes down to the second option which is Active monitoring, it is usually concerned with investigating some services or aspects of the network’s performance owing to observing some of the effects of injecting traffic into the network node that is managed or is under monitoring. Therefore, an Injected traffic emerges as appropriate to the service or aspect that is under investigation. For example using Internet Control Message Protocol (ICMP) ping packets to check or more over establish reachability of a managed node, And using HTTP requests when wanting to monitor the response time of a managed server(Landfeldt, Sookavatana, Seneviratne 2000).

Active network monitoring, releases test traffic onto the network and observes that traffic as it travels through. This traffic is not taken from actual transactions that occur on a network, but rather sent through the network in order for the monitoring solution to examine it on its path.

Test traffic usually mimics the typical network traffic that flows through a system so that we will gain the most relevant insights to its network. Because active network monitoring pumps traffic into the network, it is used to determine a network’s performance in real-time. It also does not rely on actual traffic, meaning the network team can use synthetic monitoring at any time; it does not need to capture network traffic for it to work(Shamsi, Brocmeyer 2016)

3.3 Technology Background

Most of all, the real principal background or history of Network monitoring highly depends on the history of the simple network management protocol (SNMP). The 1990’s and 1980’s marked the start of packet networks, with commodity ethernet networking for local area networks, frame relay and X.25 for wide area network. There was a need to find a common and standard way to gather monitoring statistics. During this time , the government of the United states (DARPA) was working on developing the protocols (Tom Foottit 2020).

It was in 1988 that the Simple Network Management Protocol standard was approved, bringing the base for present day Network performance and configuration management. There are various collections and information’s regarding managed nodes on the network infrastructure. Furthermore, it helps in modification in changing a node’s traits. It’s a fact that SNMP is broadly used in present day for purposes of Network management and monitoring(Mauro, Schmidt 2005).

In The 1990’s there also was a growth of data warehousing, which introduced various data back to one group space. This has a great advantage as it enabled engineers or any user to be able to access and analyze data. In addition to that these warehouse were purposed to carry or support BI and analytics, which in turn makes an important era in the revolution of collecting data for network monitoring performance, meaning organizations will be able to collect information or data from the managed network and analyze it to oversee and understand historical alerts and events which simplifies the process of troubleshooting(Tom Foottit 2020).

By the mid ’90s the commercial Internet had taken root, ushering in a new phase of networking, and bringing some improvements to network performance monitoring, data collection and analytics.

Cisco introduced NetFlow into its routers in the mid 1990’s, NetFlow is one of the various network flow standards which is even though not broadly but still is used to produce network performance data. It collects IP network traffic information and data as it gets in to and exits a port interface, even though this information and data was collected by using SNMP. This helped different network management and monitoring systems to analyze source and destination traffic and causes of congestion to pick out on malicious behavior(Held 2002).

With the growth of the internet and network infrastructures getting complex in the early 2000's the amount of data that needed to be collected for the purpose of monitoring increased exponentially. simultaneously network configurations and performance monitoring became very pivotal and got linked to business stability, productivity, and security. Despite that there are some alternative protocols getting introduced to the monitoring system world SNMP remained the most common way to gather monitoring data from a managed node(Tom Foottit 2020).

Nevertheless, SNMP protocol applies the concept of polling while trying to collect data from a managed node at some regular time intervals this in turn might start affecting the performance of the network infrastructure under monitoring. To reduce this effect performance data is collected mostly only 5 to 10 minutes making the monitoring system more reactive than proactive(Mauro, Schmidt 2005).

At the time the world of networking met the Web 2.0 technology the industry needed some changes: A better amount of performance information and data was capable of being handed over and analyzed in real time. These results are observed in Open gNMI and Open config-standards that different network vendors have begun to introduce and adopt to make it possible to stream telemetry data in near real time. Despite the fact SNMP is widely used in the monitoring world ,open config based protocols and standards has started to overtake it because of the advantages that they give a better and efficient monitoring for large amount of data (Goransson, Black, Culver 2016).

Through time the complexity and speed of networks has not slowed down to increase. In present day having the aim to lower the price of network management down, organizations and the industry in general is looks to real time data collection closed loop automation and analysis to contrivance variety of network configuration and management tasks that were used to be performed manually. A network monitoring system grabs a nearly real time data and equipment statistics from virtualized or physical infrastructure in a cloud environment as well as SDN software defined networks and then will produce and analyze recommendations from the collected data in a real time is the aim(Tom Foottit 2020).

3.4 SNMP (Simple Network Management protocol)

SNMP is a protocol that defines the monitoring data exchanges between one or multiple management systems and multiple agents or managed devices. It gives a standard framework for keeping formatting and storing monitoring information's. Furthermore, it defines various multi and general-purpose management information's and objects. At first SNMP was built for the purposes of network performance and management for networks and the internet as well operating TCP/IP. It has expanded for different uses in almost all types of network infrastructure and environments. In the industry the term SNMP is mostly used to represent the collections specifics for network configuration and management that consists of the protocol by itself, the construe of the database and other related associated concepts. In general SNMP utilizes UDP port 161 or 162(Juan 2016).

Different organizations monitor their LAN and WAN infrastructure by using SNMP. To monitor networks by using SNMP an admin will need to configure the node that needs to be monitored in other words the SNMP agent to dispatch the monitoring data to the SNMP manager. SNMP operation consists of three key parts or components which are the Managed devices, agents, and the network management station(Hare 2011).

a) SNMP manager

The SNMP manager or sometimes referred as the management station is the keystone of any network management infrastructure. It supplies the interface in between the administrator of the monitored infrastructure and the managed system in general. Mainly the task of the manager is to acquire with SNMP requests, needed information or data about the nodes connected to the network infrastructure.

The SNMP management station will have ,a user interface , sets of management applications for data analysis and the ability to translate the network manager's requirements into the actual monitoring(Michalak 2018). The SNMP manager polls end stations or nodes and their SNMP agents to verify and check the values of configured variables defined in the management information base MIB. This polling process can be automatic or it can also be user initiated either way the SNMP agent will answer to all the polls(Steinke 2003).

b) Agent

On a managed network the SNMP agent is what resides on a managed device or node. These devices can be any devices that resides on the monitored network infrastructures like hosts, routers, bridges switches firewalls, printers ,IP phones extra .These agents respond to requests made by the management station (Morris 2003).

c) Management Information Base

The management information base MIB is a compilation of information that is organized in a form of hierarchy. Generally, there exists two types of MIB which are scalar and tabular. The scalar ones define a single object instance and the tabular can define multiple related object instances that are assembled into a MIB table. For example the typical objects that might be monitored on a printer could be the number of printed files, the different cartridge states and the like, and on a switch this objects could be the outgoing and incoming traffic and the number of packets addressed to a broadcast address or the rate of packet loss and latency(Presuhn, Case, McCloghrie, Rose, Waldbusser 2002).

MIB has a way to provide a path to define managed objects and their traits. It is the list of these objects collectively that the manager can use to decide the overall health of the managed device where the agent resides. The MIB can generally be a databased of different managed objects as explained above that the agent tracks.

According to the need any sort of status or relevant statistical information that can be accessed by the manager will be defined in the MIB. The basic functions of MIB can be explained by using an example of how we use a dictionary, just like we use dictionaries to see how to spell a word and its definition or meaning, a MIB defines a textual nomenclature for a managed object and describes its meaning. A certain agent may implement multiple MIBs, but according to RFC 1231 all agents implement a MIB called MIB-II. The main aim of MIB-II is to provide a general form of the TCP/IP information for management(Mauro, Schmidt 2005).

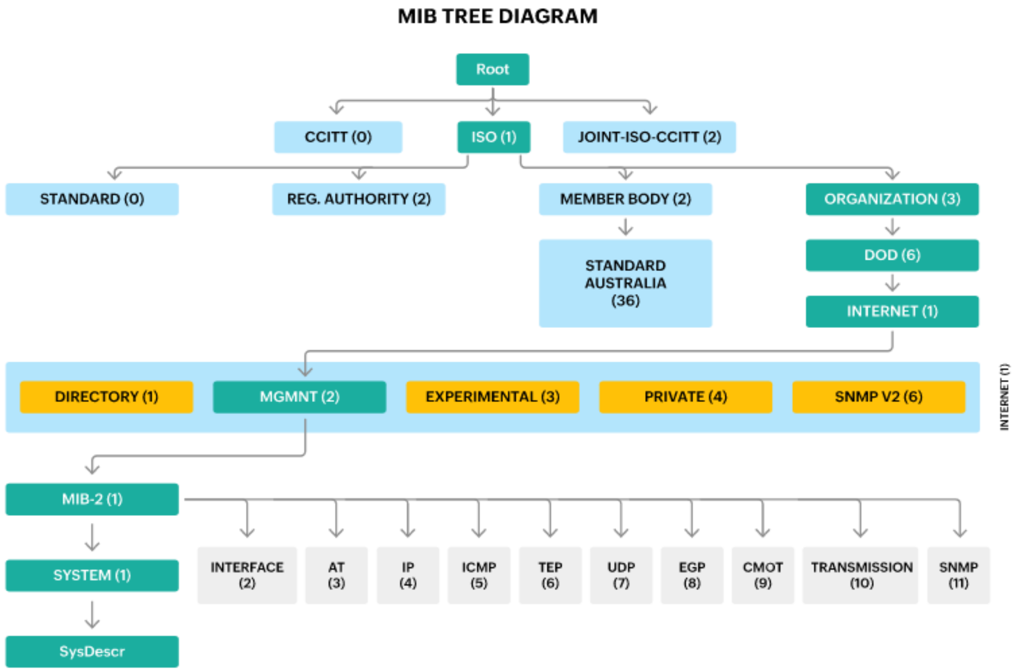


Figure 1: MIB hierarchy (Mauro, Schmidt 2005)

As shown on the above figure the managed devices that are managed by SNMP keep their management information in a form of a tree. The very first internet standard defined approximately about 120 object types. As shown in the figure under MIB hierarchy, the root at the top splits into three major branches: Consultative Committee for International Telegraph and Telephone (CCITT), International Organization for Standardization (ISO), and joint ISO/CCITT. The text strings explain the object names whereas the numbers allow software to compact, create encoded presentation of the names. In some cases, the manageable objects can have permission to read only or it can also have the permission to read and write(Hare 2011).

3.5 Basic SNMP operation

In a nutshell the agents and management station are connected by a management protocol, that has capabilities like Set, Get and Trap. In a monitored network infrastructure, the devices that are needed to be monitored or agents will store the monitoring data and responds to the SNMP manager when requested to. The other way agents communicate with SNMP manager is asynchronously by sending alerts to the SNMP manager by using special PDUs called the SNMP trap(Mauro, Schmidt 2005).

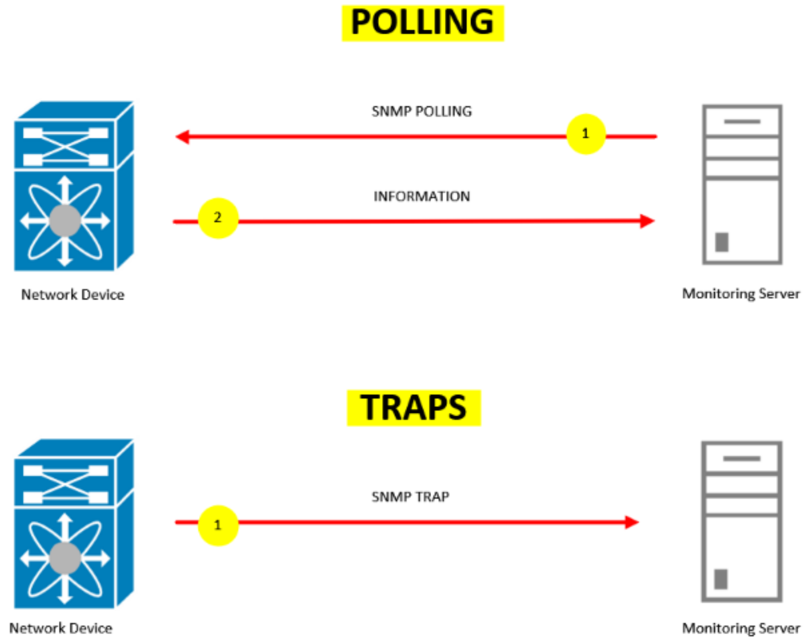


Figure 2: SNMP polls and traps process (Murray, Stalvig 2008)

SNMP uses UDP as a transport layer protocol while passing the monitoring information between the network devices or nodes and the manager. The reasons that SNMP uses UDP over TCP is that UDP is connectionless, as in there is no end-to-end connection formed between the managed nodes and the manger while communication. On one hand this makes it to be unreliable since there is no acknowledgment of data being received but it is up to the SNMP application itself to figure out if any datagram is lost between the monitoring communications.

This can be visible simply by time outs. The manager sends requests and waits for the agent to respond back the waiting time depends on the configuration if the manager did not hear from the agent in that specific time it will then assume that the datagram is lost and will request the information again(Mauro, Schmidt 2005). When it comes down to traps the situation is somehow different, at times where the agent sends out traps to the manager and it does not reach the manager then there is no way that the SNMP would know that the trap has not arrived. Worst thing the agent has no way of knowing that the information or trap it sent has not arrived and that it needs to resend it. Despite that the upside of using UDP is the fact that it requires a very low overhead, so that the load and impact on the monitored network performance is somewhat reduced. There are cases where SNMP is implemented over TCP but that only happens for special cases, In a complex network infrastructure implementing SNMP over TCP would be the worst idea(Murray, Stalvig 2008).

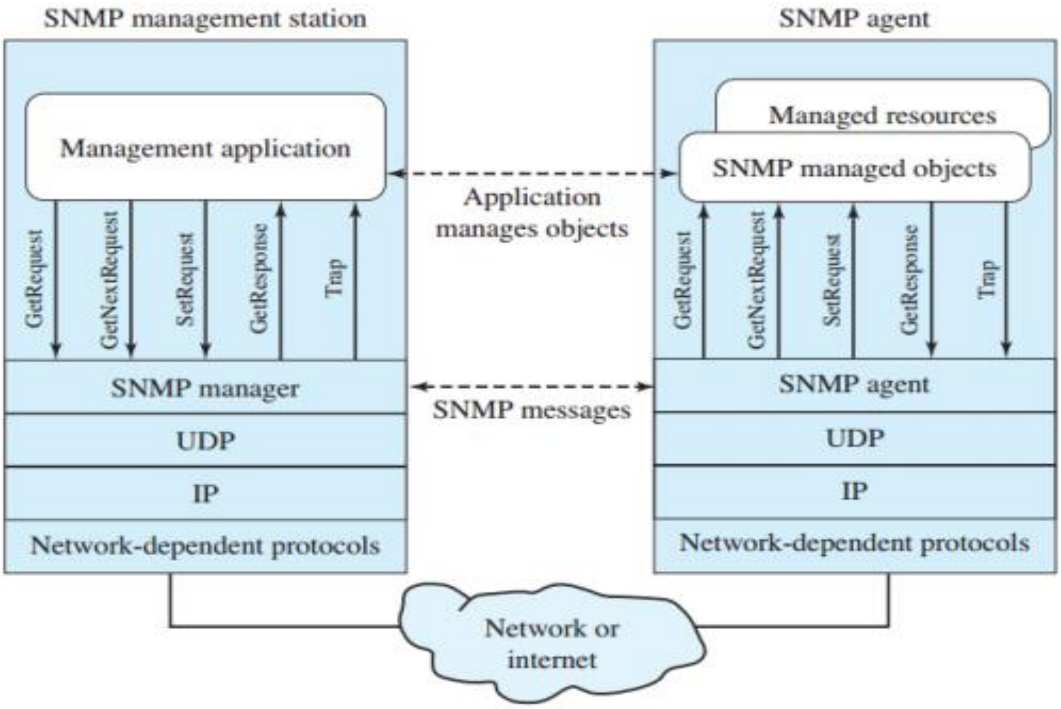


Figure 3 :SNMP function and operations(Mauro, Schmidt 2005)

On a network infrastructure that is under monitoring every node carries a management information base and an agent. the agent keeps checking the data about the managed device and prepares it to be available to the SNMP manager to transmit it when requested.

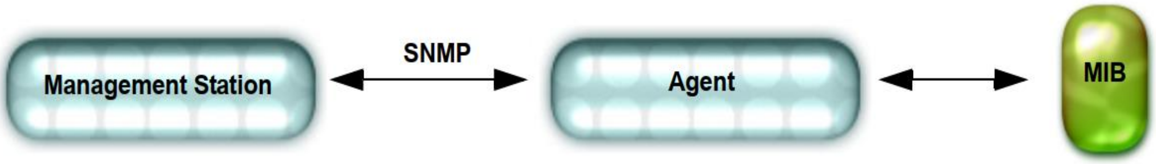


Figure 4: Basic SNMP structure(Brocade Communications Systems 2015)

When under operation the management station can SET or GET monitoring information when its inquiry an agent. there are four basic SNMP commands that are the SET, GET, getnext and get response, one the value is collected the agent will respond back to the SNMP manger. Agents make use of variables to respond and report the monitoring data as the number of packets and bytes in ad out of the managed device or the amount of broadcast communication received and sent(Michalak 2018).

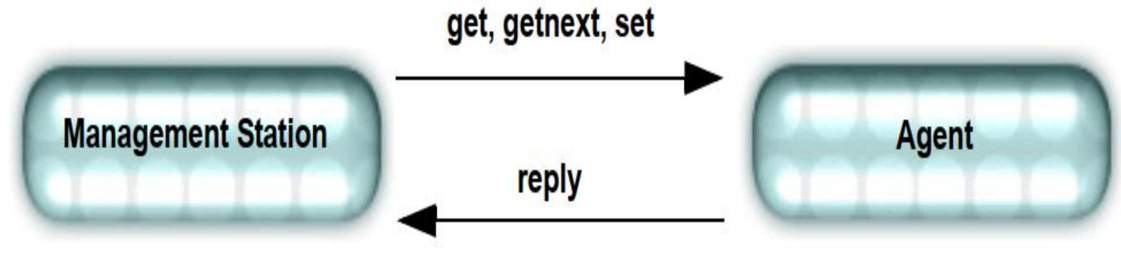


Figure 5: SNMP query(Brocade Communications Systems 2015)

In a situation where an unusual event occurs on an agent, the agent sends a trap to the management station.

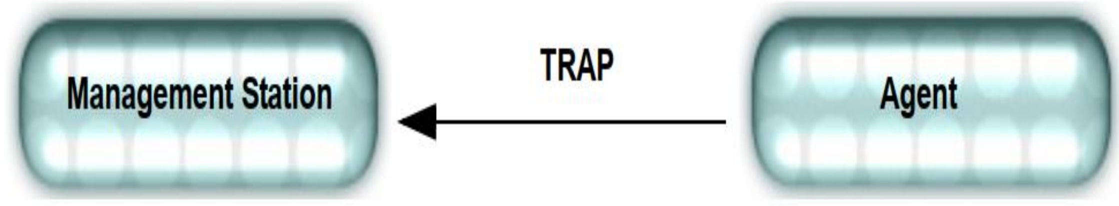


Figure 6: SNMP Trap (Brocade Communications Systems 2015)

An agent can send traps up to six different management stations, but it can receive queries from one or multiple management stations. The SNMP manager receives two types of messages, traps, and informs. The major difference between the two is that when an agent sends traps there is no way of acknowledging if it is received whereas in the case of informs the agents keeps retransmitting until the informs are received by the management station (Hare 2011). As described the SNMP agent needs to start up to be able to generate traps and send monitoring information when inquired, some statistical monitoring data may or may not be obtainable depending on what has been configured on the monitored device.

There are several factors to be considered when one decides on what must be configured and monitored for example one can consider the type of device the agent is, what model the device is, what its capabilities are extra. One also must make sure that the devices support SNMP(Brocade Communications Systems 2015) . As known not all network infrastructure and devices are not equal and configured in a similar way nor do they have similar configured features. Some environments may depend on highly on a feature and others might find similar features to be irrelevant or insignificant. Despite that there also are some features that can be found in almost all environments and are identically significant to monitor(Juan 2016).

3.6 SNMP Versions

There are three SNMP versions: SNMPv1, SNMPv2c and SNMPv3. these different versions of SNMP support different features. Their application could also be different based on complexity and scale.

a) SNMP Version 1

The initial version of the SNMP protocol is the SNMPv1 which is defined in the RFC 1157 and it also has IETF standard which is historical. With regards to security and authentication SNMPv1 uses the notion of communities to establish trusted communication between managers(Schönwälder, Marinov 2011).

These community names are primarily passwords which are almost exactly like a password someone can use to gain access to their personal computer There are three different community strings that are the read-write, read-only and trap. these strings control different kinds of activities. These community strings are plaintext strings that is danger as it allows any SNMP based application that knows the string values to obtain access to the management information of the managed device. Despite that SNMPv1 is to this day the primary SNMP implementation that is supported by many vendors. Furthermore, it is easy to set up(Rose, McCloghrie 1990).

b) SNMP Version 2

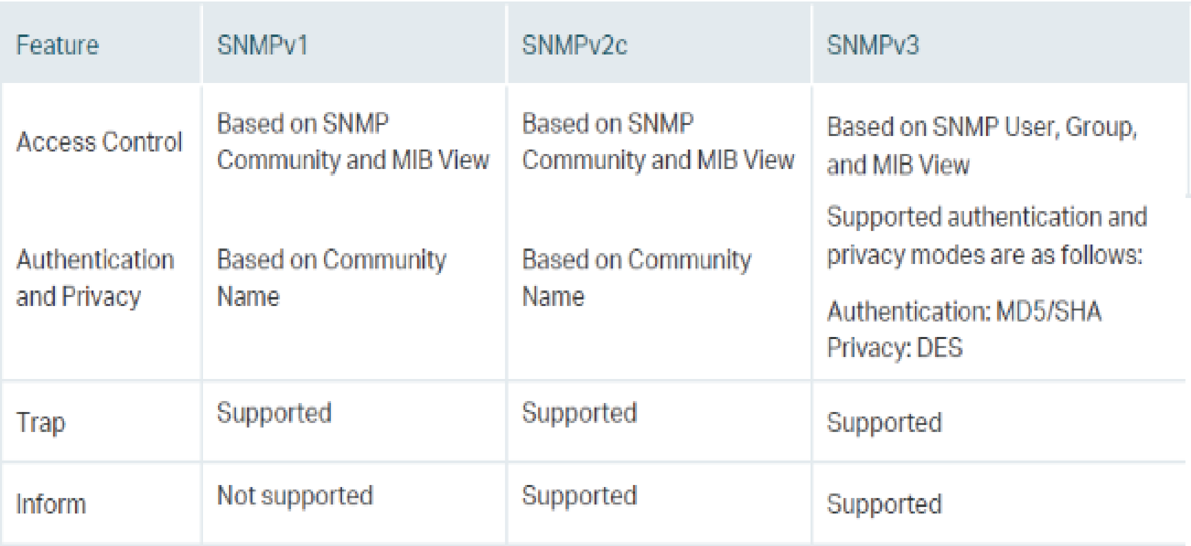
Both SNMPv1 and SNMPv2 use the application and concept of community strings but SNMPv2 is frequently referred as a community string based SNMPv2. The official and technical nomenclature of this version of SNMP is SNMPv2c. to this day there are three defined RFC for SNMPv2 the RFC 3418, RFC 3417 and RFC 3416.What makes this version of SNMP from the first version is that it adds support for 64 bits counters. The majority of different vendor devices these days support SNMPv2c automatically(Presuhn 2002).

After the first version served many years some gaps and limitations were observed like how protocols fundamentally operated, object definitions of the management base and security SNMPv2c were defined to address these limitations. Even though it addressed the limitations mentioned it also introduced numerous problems upon its introduction.

These was caused because of the security issue on the original version of SNMP, which at the end led to categorization of various SNMPv2 variants because of the multiple ways that were proposed to add security to SNMP, there was no universal agreement on implement the change. This version introduced some protocols that most people are not used to seeing in TCP/IP(Mauro, Schmidt 2005).

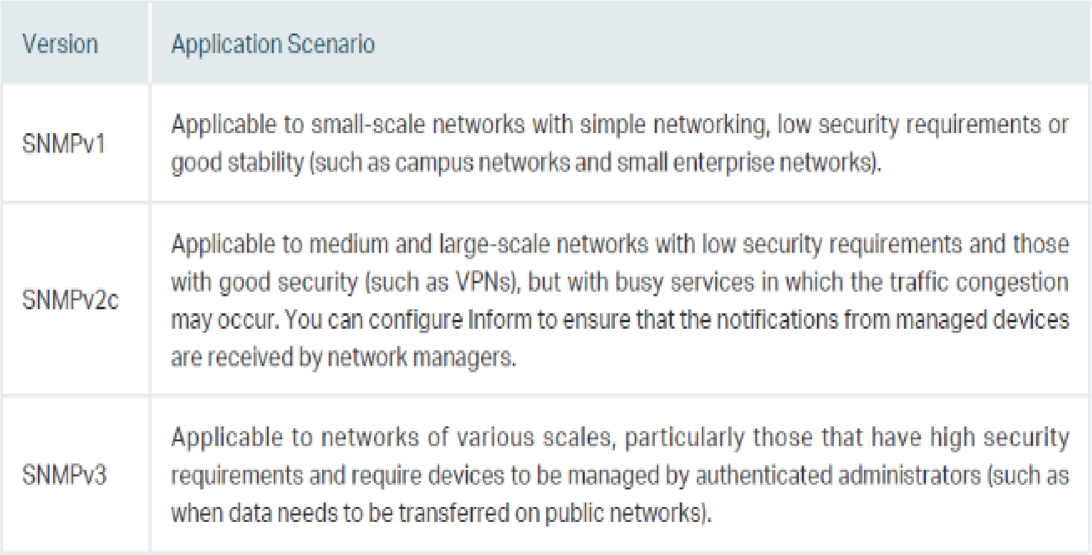
c) SNMP Version 3

The current and last version of SNMP is SNMPv3. It primarily involved the enhancement of security. unlike the previous version this version provides security with privacy and authentication, and its administration offers remote configuration and view based access control.(Hare 2011) Generally, all the three versions of SNMP share similar basic components and structure, they also follow similar architecture. SNMPv3 came with new features such as notification destinations and proxy relationships, usernames and key management, people and policies, logical coexisting, admin wise authorization and access control identities and information(Jin, Ha, Smith 2008).



Feature	SNMPv1	SNMPv2c	SNMPv3
Access Control	Based on SNMP Community and MIB View	Based on SNMP Community and MIB View	Based on SNMP User, Group, and MIB View
Authentication and Privacy	Based on Community Name	Based on Community Name	Supported authentication and privacy modes are as follows: Authentication: MD5/SHA Privacy: DES
Trap	Supported	Supported	Supported
Inform	Not supported	Supported	Supported

Figure 7: Features Supported by Different SNMP Versions(Jin, Ha, Smith 2008)



Version	Application Scenario
SNMPv1	Applicable to small-scale networks with simple networking, low security requirements or good stability (such as campus networks and small enterprise networks).
SNMPv2c	Applicable to medium and large-scale networks with low security requirements and those with good security (such as VPNs), but with busy services in which the traffic congestion may occur. You can configure Inform to ensure that the notifications from managed devices are received by network managers.
SNMPv3	Applicable to networks of various scales, particularly those that have high security requirements and require devices to be managed by authenticated administrators (such as when data needs to be transferred on public networks).

Figure 8 : Application Scenarios of Different SNMP Versions (Jin, Ha, Smith 2008)

3.7 Network key Performance Indicators

Even though there are no defined sets of universal standards, there are some common categories of network key performance indicators that are used frequently. These indicators are used to give a clearer image of the impacts and progress of different activities in a particular area of a network infrastructure. They are crucial as they can help organizations and admins measure and understand their use by examining their failure or success. The most common performance indicators are discussed below.

a) Device health

Memory allocation and CPU, memory utilization fan and temperature status are the most common KPI’s for determine a device’s health. Their values or other similar values can be gathered from almost every physical device on a network infrastructure under monitoring(Gaillard, Barthel, Theoleyre, Valois 2016).

b) Device availability

Consummately, admins want to be able to see the availability of a monitored device in almost a real time manner as it can get. The availability of a managed device is one of the most important aspect to be monitored. Unreachability of a managed device or node could cause outages on a network which may result in a high business impact and more(Gaillard, Barthel, Theoleyre, Valois 2016) .

c) Latency and packet loss

One of the most important indicators of a great connection is that its packet loss or latency is well managed and low. Being able to monitor the packet loss and latency gives a warning of problems that can say a lot about the health of the managed device on the network. Generally, if the latency of a network is increasing it affects the network negatively and admins will know that there exists a problem on the network. Same goes to packet loss, having a high packet loss on a network shows that there is something wrong going on the network under monitoring(Delsing, Eliasson, Leijon 2010).

Latency of a network is the time that is taken by a packet to travel between the destination and the sending device. the measures of latency on a network can vary for two major reasons the first one could be due to changes of route and the second one because of some congestion or delay on the network(Gaillard, Barthel, Theoleyre, Valois 2016).

d) Network interface

This KPI is one of the most important as it helps admins access the capacity of a network and be able to foretell if there is going to be a need for an upgrade or not. Just like a node an interface also gets polled to see it availability ,errors and discards per interface both for inbound and outbound (Delsing, Eliasson, Leijon 2010).

e) Congestion

Network congestion is one of the major problems that are seen on a large network. Generally, it occurs when a network is not capable to handle the amount of traffic that is passing with in it. In most cases it a temporary state rather than permanent but the effect it has on a network is significant and can cause larger issues with high business impact. Although a poor network design can cause congestion , faulty devices , over utilized devices and mis configuration can also be a cause(Welzl 2005).

3.8 General Challenges and Limitations of SNMP

a) Security

One of the crucial limitations of SNMP is its deficiency of security. The first 2 versions of SNMP use the concept of community string to check authority they almost do not have any form of authentication. Because of its initial purpose of creation which is to manage a network very little attention was given to the security feature on the first two versions. The third and last version use encryption and other different security measures that the first two versions does not have(Chatzimisios 2004). Although the last version provides some security, it barely protects from security dangers specially if it is not carefully implemented.

Potentially, there is a way which an attacker can prevent managed devices from sending traps when authentication fails. There are also multiple tools that can perform packet capturing which can lead to some attacker ear dropping to obtain some critical information about the network. The SANS institute puts the security treats of using SNMP on the top 10 internet treats. The security needs of a network can be achieved from other layers and mitigation methods for SNMP vulnerability(Schönwälder, Marinov 2011).

b) Lack of application interfaces

In terms of plots, SNMP alone does not have the ability to provide any user interfaces, visual displays and more. The application programs are not contained within the protocol itself. The applications must be built separately(Schönwälder, Marinov 2011).

c) Unreliability

As discussed, SNMP uses UDP protocol, which makes the communication to be unreliable. While polling managed devices it does not have of acknowledging transferred data, it can only depend on timeouts to understand if the monitoring data that has being transferred has been received or not. In the case of SNMP trap there no way that devices would know if the trap was dropped or successfully received by the manager(Schönwälder, Marinov 2011).

SNMP packets might also be duplicated, delayed, or corrupted. The upside of using UDP is that it has light headers and its speed there are a few special cases where organizations decide to implement SNMP over TCP but it’s not seen quite often(Murray, Stalvig 2008).

d) Incompatible trap versions

SNMP is incompatible with other protocols like DNP3 and Modbus. It is possible to obtain the compatibility via an associate SNMP conversion device. There could be a problem if agents send out nonstandard traps. There are cases where organizations decide to format their traps to meet their special needs which causes trouble eventually if or properly and carefully documented(Bibbs, Matt, Tang 2006).

e) Scalability

One of the major concerns of SNMP is its ability to scale well. SNMP devices are restricted to the volume of monitoring data they transfer, it is not ideal to transmit and handle larger quantities of monitoring information. The larger the network infrastructure gets or the more agents and managed devices the higher the bandwidth for SNMP uses. which can affect the managed network infrastructure in a negative way(Bharadwaj, Flores, Rodriguez, Long, Marai 2016).

The more device is being introduced in a network and the more SNMP interacts with it the more load that puts into the network infrastructure. Typically, these days solutions are limited approximately up to 10000 sensors. To tackle this one of the major mitigations is the applying the concept of distributed monitoring, having secondary servers report to the central server. Although this is believed to reduce the load , wen feeding data into the central server, the limitation will still remain at the central server(Michalak 2018).

f) Seclusion

The way in which SNMP's language is supported can occasionally be limiting, even though it is a protocol that is open from any language. Depending on which kind of SNMP device in monitored network, they may not fully work cohesively with other third-party SNMP devices that are provided by some other vendors, which makes it a little difficult to form a smooth connection between SNMP devices and other vital systems(Slabicki, Grochla 2016).

g) Jitter

A few articles present the notion of SNMP causing congestion on networks where there is too much load on SNMP to gather all the needed inputs and as a way of making the network more proactive. This will cause delay in the traffic of data packets over the network creating jitter. Although there are studies that conclude that the jitter SNMP can bring on a network is negligible others state that there is not enough research to conclude that SNMP is not causing any negative impact on a video or audio quality due to network jitter(QOS 2018).

3.9 Limitations of SNMP with regards to complex Networks and Large Data transfers

It is not effortlessly possible for all the versions of SNMP to be able to support the complicated OSI style data structures like complex data transfers, arrays, bulky data like large visual information and routing tables. While SNMP works well for small and medium scale networks it suffers and is not well suited for retrieving giant amount of monitoring data. The main reason for this is the way SNMP gathers its data from managed devises, a poll method. SNMP only gets information from a managed device when its requested. This by itself can hamper problem detection while troubleshooting on a large complex network(Shaffi, Al-Obaidy 2013).

Generally, SNMP cannot be considered ideal for management of complex and large-scale networks because of the performance issues that may arise from polling. Messages square measure is only set if and only if a message must be sent, meaning SNMP is asynchronous therefore there will not be any automatic way of confirming if a device is reachable or online. A polling approach can also cause a delay up to 10 minutes before the monitoring information is received from a managed device(Amirthalingam, Moorhead 1995).

For the most part in large sized networks because of lack of time stamp and network lag on the data when it gets out of the network element, the polling data does not all the time arrives at the monitoring system in the exact similar order as it was being requested from the targeted managed device. This means that the monitoring data received might have some old blips that takes time for engineers to accurately analyze and understand(Santos, Esteves, Granville 2015).

The other big problem of using SNMP on complex and large-scale networks is that it normally runs on the CPU that is allocated inside the managed device, which will affect the network by slowing the performance of the network itself. In modern systems one feature of great monitoring system is its ability to report in real time, SNMP is not that fast to be able to provide that feature. As an example, if SNMP polls a managed device every 5 minutes for its reachability the device might go done on the 4 out of 5 minutes.

When implementing SNMP on a WAN environment the response time for normal traffic and the SNMP traffic itself slows down. When SNMP polls all the nodes in the complex network , the managed devices perform extra processing , for all these reasons its better not to poll a complex network more frequently , and if not frequently polled the monitoring system gets more reactive than proactive (Shaffi, Al-Obaidy 2013). Improper usage of SNMP or configuration mistakes would result in having false alarms which wastes engineers time and would also lead to an investigation on faulty hardware configuration when the real root cause is just a poor attempt to monitor it (Mahajan, Joshi, Khajuria 2012) . in today's large-scale complex network, the environment has eventually become less homogeneous which is making it hard for SNMP to be as effective as it when it monitors small and medium homogeneous networks. Packet drops has become a frequent problem which is highly impacting visibility in complex networks. The other limitation is that to this day there is no single purpose object identifier(Hamid, Kawahara, Asami 2010).

3.10 SNMP Alternatives

To this day it is difficult to find a complete alternative or replacement since it has been implemented into hundreds of thousands and millions of software and systems for over 30 years now. There are quite a few alternatives of SNMP for server monitoring that emerged starting the mid 2000's. Microsoft has been forcing the WMI protocol for their windows servers. There are other powerful opensource agents that can work with different monitoring systems to monitor servers like Collectd, telegraf, netdata Prometheus in combination with Grafana and more(Mahajan, Joshi, Khajuria 2012).

There also some SNMP alternatives emerging for network monitoring especially with the appearance of SDN software defined networking and IOT internet of things. SDN brought the notion of API programmable interfaces for monitoring and configuration. There new standards emerging recently for networking monitoring like streaming telemetry, the gNMI, gRPC and RESTCONF protocols that can provide different methods of communication with monitored devices and flexible interfaces. Before a device can receive and/or send monitoring or configuration data over RESTCONF, NETCONF, gRPC or gNMI protocols the device must be encoding that data with a defined data model. Presently, YANG is the most famous data model and is quickly becoming an industry standard but, there are other data models from different vendors. The encoding of monitoring and configuration data is performed in a text format using JSON or XML or by using binary format protocol buffers(Santos et al., 2018).

Microsoft announced that SNMP is deprecated on windows servers back in 2012 and not more than 2 years ago in late 2018 google has published a paper with a title 'SNMP is dead' but the reality is that the 30 years old protocol SNMP keeps going for its character that its alternatives yet does not have, its light, familiar, simple to use and free making it hard to go away for good.

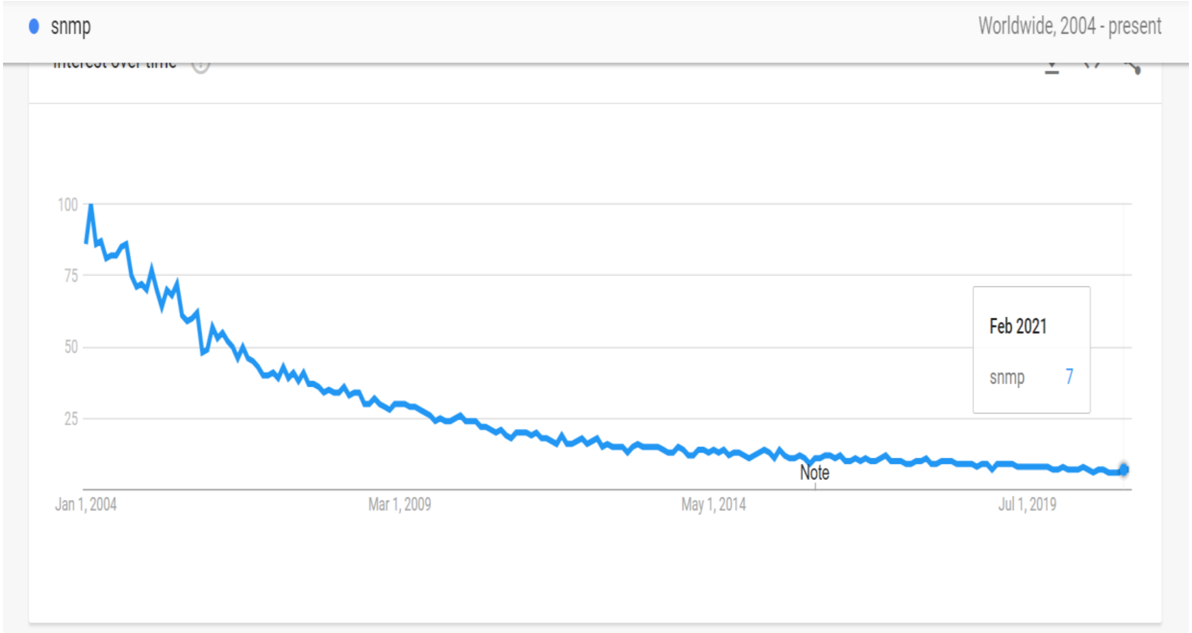


Figure 9:Picture showing interest over for the SNMP from 2004 till 2021 (based on google searches)

The above picture under figure 9 represents the interest of SNMP from 2014 up to present day based on google search results. It shows how the interest in SNMP has exponentially degraded over time. Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means there was not enough data for this term.

3.11 Related work to Comparative analysis of open-source monitoring tools

In today’s world various network monitoring tools exist from free-open-source software (FOSS) to the proprietary ones. These tools can be configured in a couple of ways first is to conduct general network monitoring and management and the second is to monitor a specific hardware component. While selecting a monitoring system majority of the time two main factors get under consideration which are the services that are needed to be monitored and the goals of monitoring them(Svoboda, Ghafir, Prenosil, others 2015). Research shows that the global market for network monitoring will increase up to 11 billion USD by 2024.A network monitoring tool might cost between 15000 to 40000 USD in their first year of usage. Today, Tens of free open source software projects are itemized under GPL, General Public License(Steiniger & Bocher, 2018).

There was an evaluation that was made in 2016 to compare different open-source network monitoring tools. Currently there are different large IT companies like HP and Oracle fabricate their own Network management system products. Nevertheless, these products are extremely expensive and mostly implemented with restrictions due to contract and license issues.

This same paper has made a detailed evaluation of three of the most used open-source monitoring tools Zabbix, Nagios and openNMS. These three tools are mostly used and are the most effective open-source tools. The evaluation was performed on a real production network environment making it more realistic and reliable(Al Shidhani, Al Maawali, Al Abri, Bourdoucen 2016).

a) OpenNMS

OpenNMS is a free opensource network management tool that was developed in 1999. It has the capacity to monitor about 70,000 devices. It can execute performance measurement, notification and alarms management ,events and service monitoring, assets management capacity and auto device discovery(He-wen 2015).The open source code of OpenNMS is based on XML and Java .Its management server can be implemented as a multi process applications, PostgreSQL database and its user interface has a multiple java server pages and multiple servlets(Gehlbach 2015).

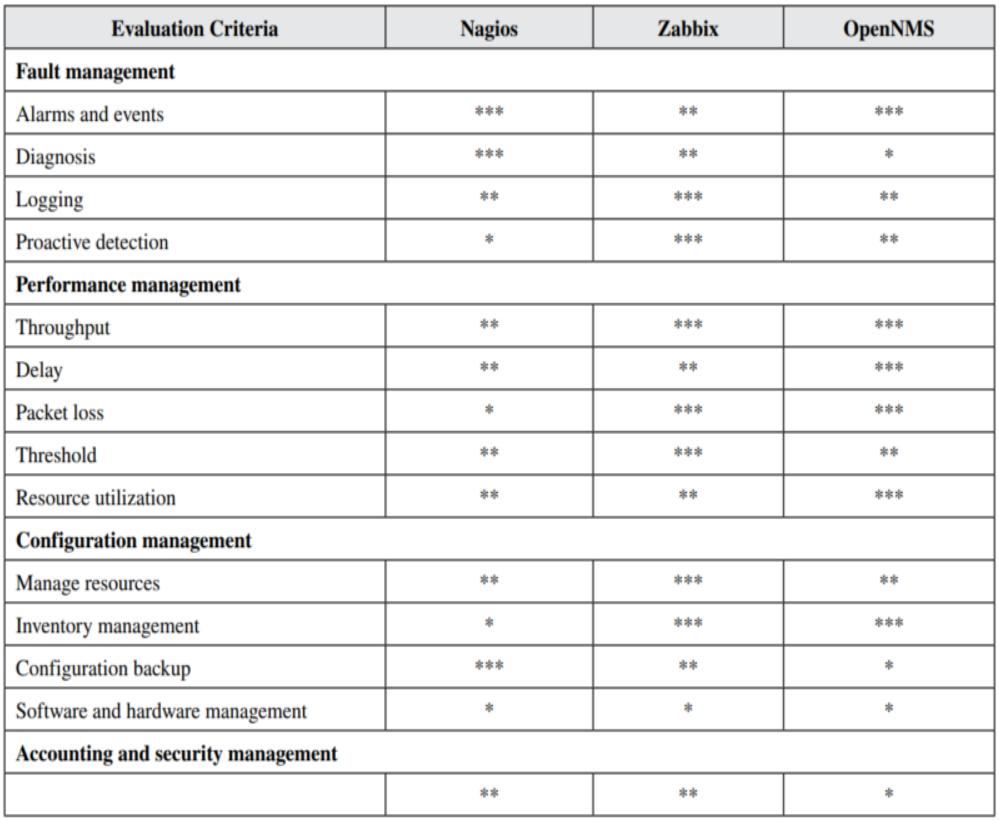
b) Zabbix

Zabbix was originally started as an internal project in 1998 but then it became an enterprise class monitoring tool. It is free and has a capacity to monitor multiple parameters of a network node apart from integrity and health of managed devices. IT has flexible ways of alerting and sending notifications, it allows admins to configure notifications for several events. Its web interface is written in PHP and the backend server is written in language C (Olups 2010). Zabbix is without issues compatible with SQLite, Oracle, PostgreSQL and MYSQL to store its monitoring and configuration data.

Furthermore, without having to install additional software it can monitor the responsiveness and availability of standard services like HTTP and SMTP. There are also Zabbix agents available , that can be easily installed on Windows and Linux devices to be able to monitor their services and resources and generates statistics like network Bandwidth utilization , CPU, diskspace and many other parameters (Olups 2016).

c) Nagios

Much like OpenNMS Nagios was also initiated in 1999. Nagios, or "Agios" transcribe the Greek word **άγιος,** which means "saint”. It is a powerful monitoring tool that has the capacity of assisting network admins to quickly identify and resolve different network related issues. It offers monitoring to applications, switches, servers, different applications and more. What makes Nagios special is its capacity to be flexible to be customized. There are also multiple solution providers to issues related to Nagios tool itself (Enterprises 2017).There are several probing sensors and monitoring extensions that can interoperate with Nagios effectively. It has another version called Nagios XI which is not an open-source tool but gives more features and support than the original tool that was developed in 1999(Gamalielsson, Lundell, Lings 2010).



Evaluation Criteria	Nagios	Zabbix	OpenNMS
Fault management			
Alarms and events	***	**	***
Diagnosis	***	**	*
Logging	**	***	**
Proactive detection	*	***	**
Performance management			
Throughput	**	***	***
Delay	**	**	***
Packet loss	*	***	***
Threshold	**	***	**
Resource utilization	**	**	***
Configuration management			
Manage resources	**	***	**
Inventory management	*	***	***
Configuration backup	***	**	*
Software and hardware management	*	*	*
Accounting and security management			
	**	**	*

Figure 10:Comparative analysis between Nagios , Zabbix and openNMS (Al Shidhani, Al Maawali, Al Abri, Bourdoucen 2016)

The evaluation that was performed on these three open-source monitoring tools was based on FCAPS functional areas. SNMP packets compatibility was one of the selected features during the evaluation. The study shows that some tools excel in some of the areas and others come short in some other areas. The study found out that all three tools are generally adequate with regards to compatibility with SNMP packets. However, OpenNMS SNMP packets compatibility exceeds the rest two in almost non-significant manner. OpenNMS also predefines configurations which in turn helps admins reduce their effort and time. OpenNMS does not offer agent modeling and has very limited flexibility for customizations purposes. Zabbix and Nagios on the other hand provide multiple agents modeling. Compared to Nagios and OpenNMS, Zabbix has shown its capability in diagnosis model to trace root cause of identified issues and network failure but when it comes to flexibility Nagios stands out of all(Al Shidhani, Al Maawali, Al Abri, Bourdoucen 2016).

4 Practical Part

The main objective as defined on the thesis is to design a monitoring tool/Approach for highly complex network systems that can resolve some of the limitations and problems that may arise by using the protocol-based approach (SNMP), to achieve this, prior to proposing the design the thesis conducts an evaluation on SNMP with an aim of spotting out its limitations and problems. Therefore, the practical of the thesis has two main sections.

The evaluation and the design, the evaluation follows two approaches, evaluation to observe if the protocol is resource intensive and evaluation on the performance of the protocol itself. Consequently, the practical part will start off by evaluating the use of applying a protocol-based approach on a large-scale network to see if using a protocol-based approach affects the network it monitors or in other words checks if SNMP uses resources intensively and affects the performance of the network by considering two Network performance indicators (CPU and Bandwidth Utilization). As a part of evaluating the performance of SNMP itself an evaluation will be conducted on the performance of SNMP with regards to how fast or slow it reports Network issues (SNMP Delay).

There are a few articles, blogs, and journals written with regards to the performance of SNMP. They tend to look like they are impartial in their conclusions and comparisons. Unfortunately, almost all of them were performed in a lab environment and they often aimed at proposing an expensive new emerging tool or approach which from a performance point of view, is supposed to outperform SNMP. However, the evaluation performed on this thesis is performed on a real production network to acquire the most realistic results.

The infrastructure under evaluation is a large-scale complex network of company X under monitoring with pure SNMPv3 over SSH. It has over 500 sites (Plants and branch offices) in 4 main regions of different continents having 2 datacenters per region connected over a backbone network. These sites have eWAN configuration which has two INET or two MPLS circuits each. BGP is the protocol that runs between routers on different site locations and there is a traffic shaping active on DMVPN INET and MPLS uplinks.

The results of the evaluation will be particularly considered during the second section which will be to propose a design of a monitoring tool that can resolve some of the problems that could arise by using a protocol-based approach on a large-scale network. In this section the thesis will propose a conceptual architecture mainly focusing on the way monitoring data is collected from a managed device to address the limitations from the evaluations conducted with defined working logic and uses UML modeling and Data flow diagrams by using visual Paradigm to graphically represent the flow of data in the new proposed design.

4.1 Methods and Preparations for Evaluations

As mentioned already the experiment was performed on a production network. Eight different nodes were selected to perform the evaluation. There was no defined selection procedure or criteria while selecting the devices for the fact that they are all in the same monitored network and there will not be any significant change on selecting different nodes than the ones already selected. As briefly shown under literature review of the thesis there are quite several network key performance indicators.

In the experiment, the thesis focusses only on CPU utilization and Band width utilization of the Protocol for two main reasons. one is that these two KPI's are the most used performance indicators and the second is that they directly and indirectly affect other performance indicators, which helps the thesis to cover larger area of experiment, gives wider ideas from different angles to be considered while designing the monitoring tool and while interpreting results of experiment and making conclusions. The experiment for CPU utilizations is performed separately for SNMP polls and Traps. While the experiment for the bandwidth is performed based on the RFC 1757 Ethernet-Utilization Formulas.

The other major part of this evaluation is also the evaluation that is conducted on SNMP delay, which is different from the evaluation performed to observe its utilization as this is carried out to evaluate the performance of the protocol itself. Details of each procedures and steps taken for the experiments will be explained briefly.

4.1.1 Evaluation of SNMP poll and Traps CPU utilization

The evaluation on CPU utilization of SNMPv3 will be performed on the eight selected nodes as explained under practical part description. The experiment will show the amount of CPU resource that SNMPv3 poll and trap consumes on the infrastructure and then observes if there exists a performance issue on the monitored network because of this consumption in an aim to evaluate if SNMP negatively affects the Performance of the infrastructure it Monitors.

4.1.1.1 Details of Procedures For evaluation of SNMP Poll CPU Measurement

The infrastructure under study is monitored with pure SNMPv3 over SSH. There is no tool or system used to measure the CPU utilization instead different commands were issued to obtain the desired measurement result. the routers under evaluation are models of ISR43/44XX, and ASR1001/1002 X both models have the IOS(XE) version of 16.6.8 from cisco technologies These routers on average costs about 450,000 to 540,000CZK each in today's market which have modular interfaces with various connection options, they have the capacity to support diverse and several access links like Ten-Gig, Gigabit, Serial, xDSL T1/E1 and T3/E3. These models are commonly used in large scale network for their capacity and security capabilities. The switches are models of catalyst 3650/3850, catalyst 2960X and C9200L-24T-4G with IOS(XE) version of 16.6.5/8.Thsese switches on average cost about 35,000 to 70,000 CZK each in today's market has about 88 Gbps switching capacity with 24X 10/100/100 links. These switches are LAN base and overly common in large scale networks. The detailed methods and procedures followed to measure and examine the CPU utilization are shown on the below test sequences.

1. The monitoring system for the infrastructure under evaluation is configured in a way to poll the nodes every 5 and/or 10 minutes, Therefore the first step is to observe these polling tags, which can simply be accomplished by going through the monitoring configuration or alternatively by issuing the command 'Show snmp stat oid' which will show the time tags and intervals of the polling request for different OID's.
2. Based on the observation from step one issue the command 'show processes CPU | exclude 0.00%__0.00%__0.00%' on the devices under evaluation.
3. Take the measurement of the CPU utilization of SNMP from the output results under the second step to see the amount of the CPU resource SNMP is utilizing.
4. Under the condition that a high utilization is observed Check logs and status of the OID which will display the timestamp and order to identify the OIDs that did respond slowly causing high CPU utilization, having this identified and checking logs will help while observing and understanding if network issues were caused on the network for that specific timestamp. Measurement results from third step can also be compared to a cisco standard to label a measurement as Intensive or not.
5. After performing the above procedures, the final step will be to evaluate the measurement result if it was resource intensive or not, see if the polling process has caused a performance issue on the network and to take notes and consideration for the proposal of the design of the monitoring tool.

The above procedure is performed ten (10) times for each device under Evaluation, meaning there will be ten samples of utilization taken from each device. An alternative method to accomplish the first and fourth steps would be to debug snmp which can inspect which Object ID were being requested at the time of the high utilization that was observed. However, issuing a debug over a production network would submerge the device under evaluation which might cause other issues. Moreover, it will make it hard to decide if performance issues were caused solely by the polling process as a debug overwhelms the devices under evaluation (Routers & switches). For better visuality of results, the ten samples taken from each device will be used to form a graph, these graphs are built manually by giving the number of times the test was performed (1 to 10) as the independent variable and 0.0 to 1 [0% and 100% when multiplied by 100]   as the dependent variable which represents the measurement results or the amount of the CPU utilization on a web-based tool called Rapid-Tables graph.

4.1.1.2 Details of Procedures For evaluation of SNMP Trap CPU Measurement

There are a few differences in evaluation procedures for Trap than the procedures performed to observe results of polls. For a managed device to send a trap out there must be an incident occurring that triggers the devices to do so. The below procedures were performed in a sequential manner.

1. First step would be to wait for the right time for devices under evaluation to send traps. To avoid time gaps most of the evaluation was conducted on a flapping monitored virtual interfaces.
2. Based on the observation from step one issue the command 'show processes CPU | exclude 0.00%__0.00%__0.00%' on the devices under evaluation.
3. Cross check the time stamps on the logs for that specific interface that went down with the timestamps from the second step.
4. Take the measurement of the CPU utilization of SNMP from the output results under the second step to see the amount of the CPU resource SNMP is utilizing.
5. Under the condition that high utilization is observed, check for logs and issues for that specific timestamp if any exists. Measurement results from third step can also be compared to a cisco standard to label a measurement as Intensive (High)or not.
6. After performing the above procedures, the final step will be to evaluate the measurement result if it was resource intensive or not, see if the Trap process has caused a performance issue on the network and to take notes and consideration for the proposal of the design of the monitoring tool.

Just like conducted for the polling process these procedures are conduced ten (10) times for each device under evaluation. The major difference from the polling process is that the trap procedure is time sensitive. With the provision that the third procedure have a significant time gap the measurement result should be discarded and performed again until it satisfies the conditions. An alternative way to perform the first procedure would be to continuously bounce a monitored virtual or physical interface under monitoring. These options were not appropriate to be used as a procedure for the fact that they would cause high business impact, instead some samples besides to flapping interfaces were conducted during a time of a change request which affects the virtual interfaces under evaluation. Similarly, as that of the polling process a manual graph of the results were built using the exact same web tool (Rapid-Tables graph) for visualization purposes.

4.1.2 Evaluations of SNMP Bandwidth utilization

These evaluations of SNMPv3 bandwidth were performed based on the RFC 1757 Ethernet-Utilization Formulas similar results can be obtained simply by using online Bandwidth utilization calculators. The conducted results of measurement can show how much of the bandwidth that SNMP utilizes. It is known that ICMP and SNMP traffic are negligible or insignificant, but this evaluation is performed to see if it uses the bandwidth resource intensively.

The conclusion that could be made from the results could only be the proportionality that the larger the sale of network gets the amount of consumed band width increases exponentially and that could be important to consider while choosing an approach to be used in the conceptual design of the monitoring tool as bandwidth comes with cost and should be well managed. Response times on telnet and packet drops will be checked for the monitoring process in general.

$$Utilization = 100 * \frac{(\Delta_{ifInOctets}+\Delta_{ifOutoctets}) * 8}{(number\ of\ seconds\ in\Delta) * ifspeed}$$

Where Δ represents the change in octets of ifin and ifout octets, the denominator represents the difference in seconds between two polls. Eight samples were taken with two polls and the time difference between the polls for each sample.

4.1.3 Evaluations of SNMP Delay

This evaluation is different from the above conducted evaluations as it follows a different approach. It is conducted to evaluate the performance of SNMP itself with regards to how fast it reports issues whereas the rest evaluations were performed to observe if the protocol in anyway has a negative effect on the network it monitors moreover to observe if it uses the network resource intensively. This evaluation is crucial and most important as it affects the response time to identify and recover issues, increases business impacts when large delay occurs, affects network downtimes and at times can cause false alarms.

The response time of SNMP was conducted on eight devices by following a simple procedure of observing the timestamp of the issue that cooccurred and the time when the notification was received during a pre known change on the network that affects some monitored services. The resulting difference between these two can show if there was any delay.

4.1.4 Measurement Results of Conducted SNMP Evaluations

The below table shows measurement result of the conducted evaluation. It shows the separate results of all the samples and their average as well.

Table 1:SNMPv3 CPU and Bandwidth Measurement Results (Author, 2021).

N0	Device and technology	CPU poll results per sample (%)	CPU poll average results (%)	CPU Trap Results per sample (%)	CPU Trap average results (%)	Bandwidth calculated results (%)
1	Router 1 (Cisco ONE ISR 4451 (4GE,3NIM,2SM,8G FLASH,4G)	(0.35,0.38 ,0.36,0.36, 0.33,0.34, 0.34,0.38)	0.36	(0.64,0.66 ,0.66,0.68, 0.69,0.66 0.59,0.64)	0.66	0.22
2	Router 2 Cisco ONE ISR 4431 (4GE,3NIM,8G FLASH,4G)	(0.75,0.70 ,0.80,0.84, 0.84,0.80 0.65,0.87)	0.79	(0.43,0.51 ,0.47,0.40, 0.53,0.30 0.36,0.78)	0.47	0.33

3	Router 3 (Cisco ONE ISR 4321 (2GE,2NIM,4G FLASH,4G)	(0.57,0.59 ,0.62,0.63, 0.61,0.71 0.53,0.59)	0.61	(0.57,0.59 ,0.60,0.61, 0.55,0.47 0.62,0.49)	0.55	0.41
4	Router 4 (Cisco ONE ISR 4451 (4GE,3NIM,2SM,8G FLASH,4G)	(0.44,0.46 ,0.37,0.31, 0.32,0.35 0.31,0.43)	0.37	(0.75,0.77 ,0.87,0.75, 0.79,0.79 0.75,0.72)	0.77	0.28
5	Switch1 (CATALYST 2960-X 24 GIGE, 4 X 1G SFP, LAN)	(0.48,0.46 ,0.56,0.53, 0.46,0.45 0.48,0.45)	0.48	(0.54,0.55 ,0.52,0.51, 0.48,0.51 0.55,0.55)	0.53	0.38
6	Switch 2 (CATALYST 2960-X 24 GIGE, 4 X 1G SFP, LAN)	(0.34,0.34 ,0.35,0.37, 0.37,0.34 0.35,0.36)	0.35	(0.55,0.62 ,0.63,0.65, 0.57,0.64 0.63,0.61)	0.62	0.23
7	Switch3 (CATALYST 2960-X 24 GIGE, 4 X 1G SFP, LAN)	(0.28,0.25 ,0.27,0.29, 0.32,0.28 0.31,0.34)	0.29	(0.54,0.55 ,0.55,0.64, 0.60,0.55 0.56,0.54)	0.57	0.29
8	Switch4 (CATALYST 2960-X 24 GIGE, 4 X 1G SFP, LAN)	(0.54,0.45 ,0.40,0.43, 0.42,0.41 0.45,0.40)	0.43	(0.56,0.53 ,0.53,0.55, 0.55,0.55 0.57,0.52)	0.54	0.21

As explained earlier for better visuality of results, the ten samples taken from each device will be used to form a graph, these graphs are built manually by giving the number of times the test was performed (1 to 10) as the independent variable and 0.0 to 1 [0% and 100%]when multiplied by 100 as the dependent variable which represents the measurement results or the amount of the CPU utilization on a web-based tool called Rapid-Tables graph.

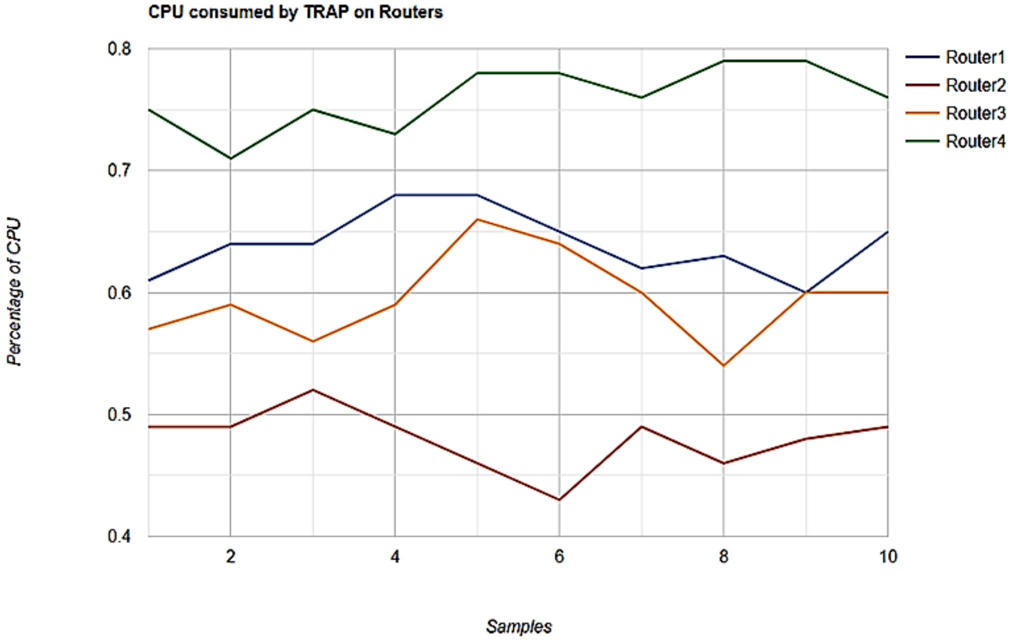


Figure 11: CPU consumed by traps on routers (Author, 2021).

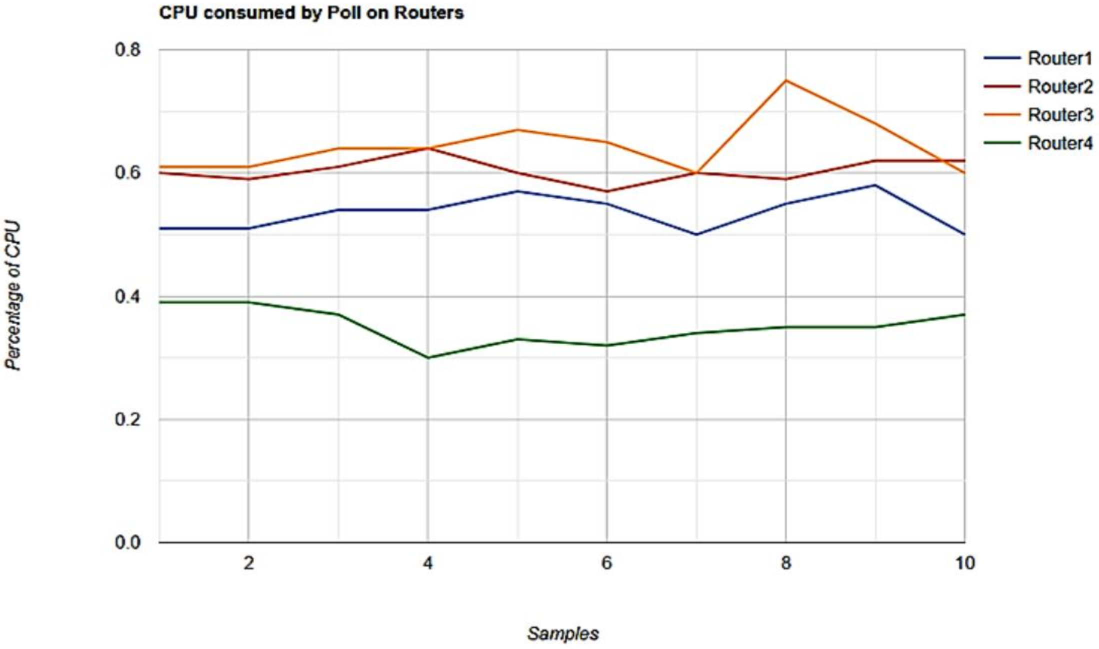


Figure 12: CPU consumed by Poll routers (Author, 2021).

According to some vendors CPU usage percentages for SNMP are considered to be a low priority process. The measurement results shows that this process consumes up to a spike of 0.8 in the poll and trap process. The polls on the routers have shown a closer result than the traps on the right side. Compared to CPU utilization standards of different vendors [0.40-0.70] the results show moderately intensive.

The measurement results shows that the routers somehow suffered from high CPU for a few samples taking too much time to process a certain request. These routers have a very large routing table being polled as they are requested for their whole routing tables every time, they learn it. The other trend observed is that the routers with the largest tables has shown higher CPU on the SNMP engine, implying that the larger the table the more intensive the engine CPU usage gets.

One of the most common character of a large complex network is having huge routing tables which can be an issue. However, as discussed according to some vendors SNMP should be a low priority processes and other process requiring CPU resource should take priority. Despite that having CPU spikes can slow down telnet /SSH response which was observed on one of the four routers under evaluation.

Other symptoms of intensive utilization of CPU resource by SNMP could be a slow ping response, routers struggling to send updates to their neighbors and a slow response on console which were not seen on the evaluation but are common symptoms and could be seen on a large, monitored network.

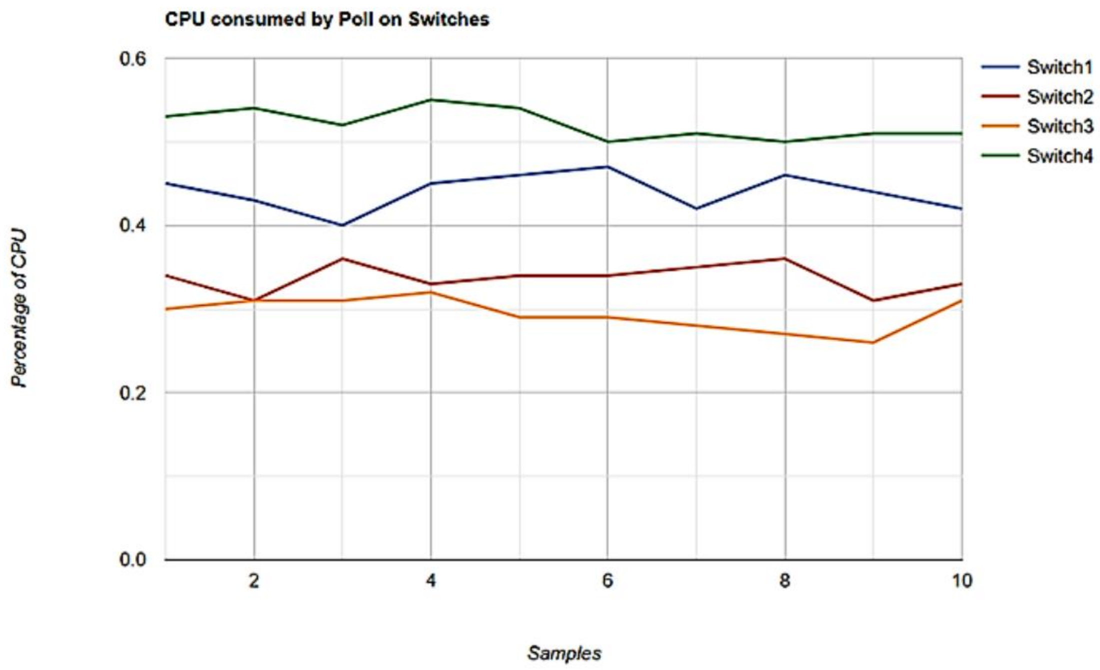


Figure 13: CPU consumed by Poll on switches (Author, 2021).

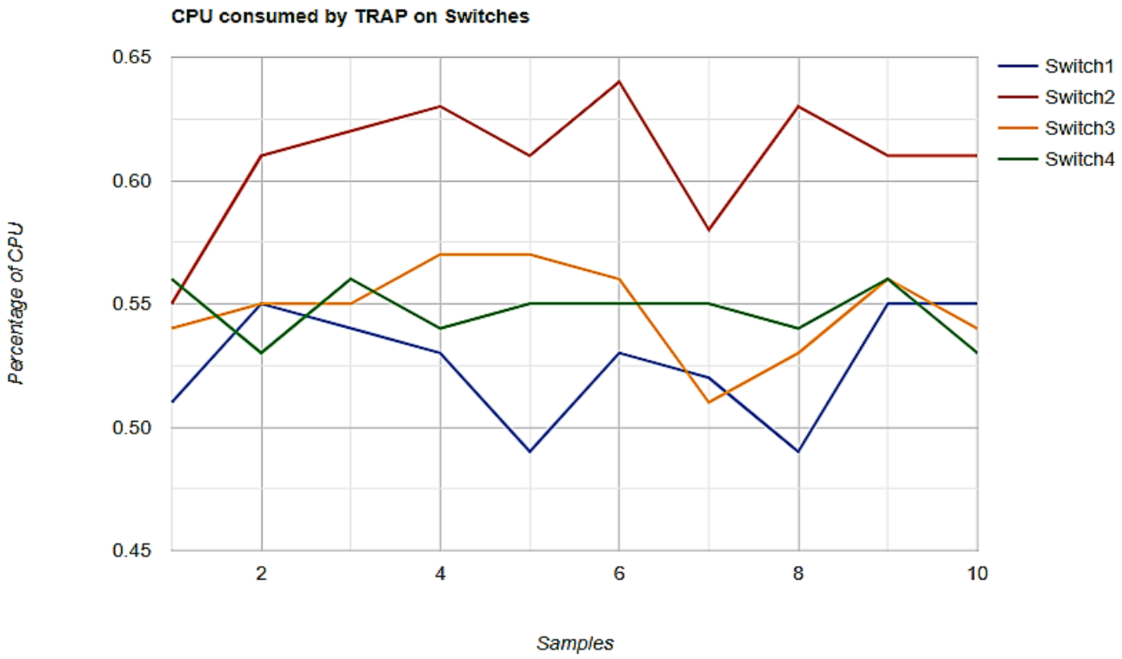


Figure 14 :CPU consumed by Poll and traps on switches (Author, 2021).

Compared to the routers the poll and trap measurement results were relatively lower. A similarity observed is the inconsistency of the trap measurement results in both routers and switches. This could be due to the difference in traps sent, some consumed high and other consumed relatively lower CPU. It was observed that the switches with higher number of VLAN IDs showed a relatively higher utilization in polls results.

Unlike the routers no symptoms of performance issues related to SNMP's engine CPU resource consumption was seen. Even though this evaluation on switches imply that SNMP does not affect the performance of the network it monitors, it cannot be ruled out that its safe as this result could change if larger number of nodes were included for the evaluation.

The bandwidth utilization on the other hand is lower on all devices under evaluation, having higher band width utilization could cause packets getting dropped, high buffer failures, latency, slow performance and more. Results imply the resource utilization is low moreover no symptoms of high band width utilization were observed. There could also be numerous factors that can cause similar symptoms because of high bandwidth utilization and to Blame this on SNMP alone would be unfair, unreal and would need in depth troubleshooting. Tools like solar winds show the separate bandwidth utilization of the protocol which helps in troubleshooting related issues. However, as for the evaluation conducted on this thesis bandwidth resource utilization is low moreover no symptoms of high band width utilization were observed.

The below table shows the evaluation outcomes of the performance of SNMP with regards to how fast it reports issues.

Table 2: SNMPv3 Reporting time and Delays (Author, 2021).

N0	Issue	Time of Notification	Time on logs	Delay
1	A failure on a port or interface is the root cause.	04:38 PM GMT	04:31 PM GMT	07 Minutes

2	The physical connection between two network adapters is down.	01:02 AM GMT	No logs/No issues	-
3	The state of the power supply for the system is not NORMAL (component is showing WARNING/CRITICAL/ SHUTDOWN.)	06:21 AM GMT	06:24 AM GMT	04 Minutes
4	Net Sweep detected on 10.x.x.x	10:42 AM GMT	10:46 AM GMT	04 Minutes
5	RX power showing low (Threshold violation)	00:33 PM GMT	00:45 PM GMT	12 Minutes
6	The physical connection between two network adapters is down	01:14 AM GMT	01:17 AM GMT	3 Minutes
7	Interface state changed to operationally down	09:53 PM GMT	09:55 PM GMT	2 Minutes
8	Tunnel interface is unstable	01:27 AM GMT	01:36 AM GMT	9 Minutes

The measurement results shows that there is a significant delay. Although the significance of the delay depends on the impact it causes, having a more reactive approach with this much delay is generally not a good sign.

There are possibilities were the issues recover by the time notifications were received. these delays will also increase the response time to identify issues, troubleshot and restore them. As seen in second raw there are cases where no issues occur on devices, but SNMP still reports them (No trap acknowledgement) causing false alarms, or issues might have existed, but logs cleared out before getting on the device just to see service is restored.

These delays would have unpleasant consequences for security nodes like a standalone Firewall. The below table shows over all outcomes and effects of evaluations conducted.

Table 3:Over all outcomes and effects of evaluations (Author, 2021).

No	Indicators	Effects seen on resource	Treats to evaluation
1	CPU	Slow telnet /SSH response was seen on routers. No effects on switches	No treats
2	Band width	No effects	If effects were seen it will not be easy to justify that it was solely caused by SNMP
3	Delay	All evaluation results show a late response except one that resulted in a false alarm where logs have cleared before Notification (Reporting)	No treats

There are several factors that can be considered to label SNMP as a less efficient protocol or not to monitor large scale networks. Although in theory It is believed that the protocol might not be as efficient as its recent alternatives, it is not a smooth and easy process to evaluate and decide if the protocol affects the network it monitors as there are numerous factors that must be evaluated and studied which are not included in this thesis.

On top of that there are times when syslog does not give the complete message and history that helps for troubleshooting and analyze previous issues or events.

4.2 Designing a performance Monitoring tool (Approach)

One of the most important characteristics of a well-designed monitoring tool is the process or methodology it uses for collecting and using monitoring data. Purposes of these collected data, data types and the frequency of data collection are also other characteristic that should be considered, but most of all the methodology is the most principal character. Hence the design of the tool majorly focusses on which methodology and process to use. while designing the tool, Besides the results of the conducted evaluation other significant factors will be contemplated.

As per the results of evaluations conducted on this thesis, using a protocol-based approach to monitor large scale networks did not show its cons in the most convincing way. This could be due to the number of nodes included under study or the lack of factors and KPI's picked for evaluation. When considering ease of troubleshooting, and not entirely depending on syslog for looking into previous events its more appropriate to consider alternatives like Netconf / Restconf and others as they do not force engineers to log on devices CLI which is hard to operationalize. Furthermore, these alternatives use a push operation than pull which can be good for performance even though it needs to be studied and evaluated. These options also give another important quality which is to enable a tool to become more proactive than reactive.

These alternatives are clearly at their beginning stage and needs the activeness and participation of vendors to take over but in an undebatable way they are the future of network monitoring and management. However, SNMP has been in the world of monitoring for over 30 years now, it is implemented in over millions of systems and software and its simplicity and familiarity with Network engineers is what makes it stay with its associated problems and limitations. Moreover, results of evaluation also did not show worst results except for the protocol's lag in reporting issues combined with other systems.

Therefore, to come up with a well-designed tool that has taken the conducted evaluation and discussed factors under consideration, the best way would be to find a way to take advantage of both methodologies in the new design, meaning the conceptual design and architecture of the tool should give the flexibility of choosing which methodology the user wants to implement in the network infrastructure.

There are multiple monitoring tools in a market that can work well and efficiently under an environment of a quite simple topology. When it comes down to monitoring complex high-speed networks with large number of nodes there are also multiple highly priced monitoring tools in the market. One of the ultimate goals of this design is to design a lower priced or no cost tool, provide flexibility of methodology in the best possible way and most of all contribute to the revolutionary transition from the legacy SNMP to telemetry options for network monitoring and management.

4.2.1 Working Logic

This section gives in depth working logic of the tool, proposes an architecture and gives descriptions of its components. The core logic behind the design is that it treats network devices and their services separately. It bases its basic monitoring by checking availability of services prior to the devices that provides them. It is completely impossible to acquire or use services when the devices that are proving those services are down or unreachable. It is always true that if service is available, the device providing it is up and running.

There is a condition seen in some tools where some nodes are intentionally configured to ignore ICMP requests and using a PING in this case to check the node’s availability shows services as reachable when the nodes providing or supporting them are down, which may look like a miracle but only happening because the nodes are ignoring the request. nodes that ignore ICMP requests usually respond to TCP SYN packet. By using this advantage, the design sticks to the logic of checking services prior to the corresponding node.

This is done to reduce the number of alarms the tool has to fire for checking every node even though no issues are observed. Having all nodes and services checked for quality of monitoring can only be advantageous while monitoring small scale networks. The design works in a hierarchical manner. Nodes are used to refer to any network device that needs to be monitored and services represent the resource that runs on nodes.

The below figure is added as an example for demonstration and explanation of the overall working logic and process. As an example, in the above figure, the starting point of check to monitor DNS on proxy is to regularly check the service. Having checked every node is not beneficiary, if service is reached there is no reason to check node 2 or node 1 and proxy for the fact that service would have not been reached if any of these nodes are not running.

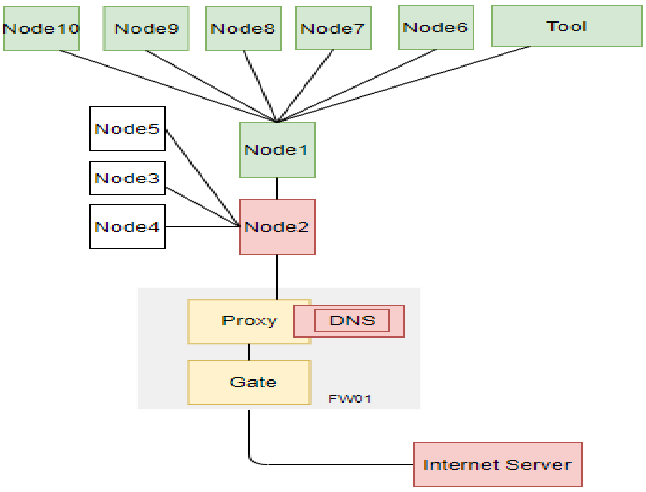


Figure 15:General working logic (Author, 2021).

Under the condition that this service fails, first thing will be to check the node providing the service, if running properly it is then the service with problems that needs to be reported, if not running next step will be to check the next connected node in this case node2 this process continuous until it gets to a node which is up and running. once this process is done the labeling of status will be performed. The below flow chart shows a general procedure of how the tool performs checks.

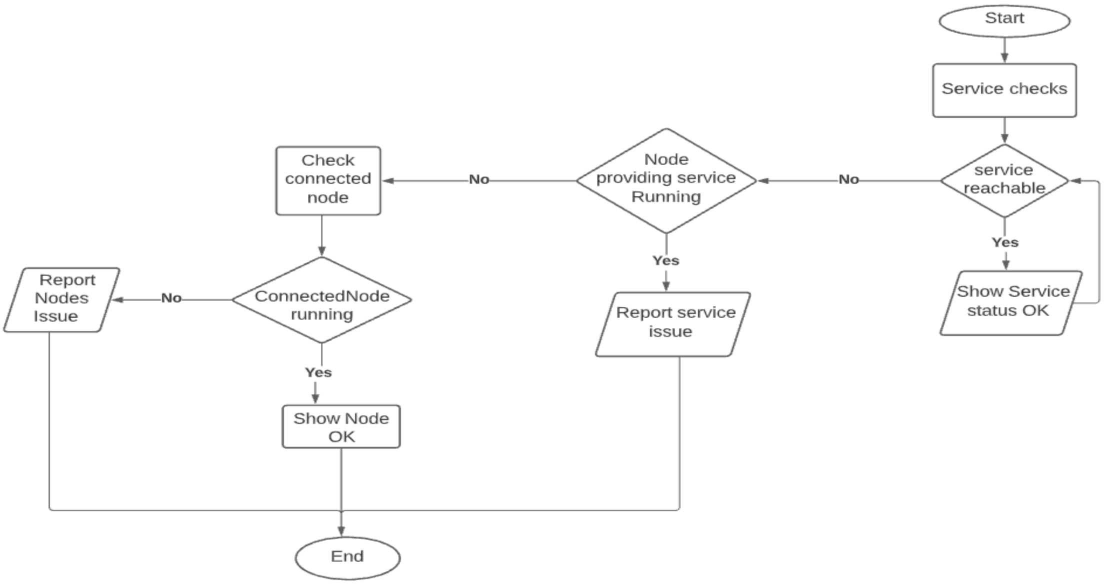


Figure 16:General working flow chart (Author, 2021).

The overall process for monitoring as shown highly depends on hierarchical relationships. There are two possibilities that the status of a node can be checked, these are either the service its providing is not reachable, or its connected neighbor (Connected node) is not reachable. Other than these two conditions nodes will not be checked which will help reduce load on the tool server especially when monitoring larger number of nodes.

In large scale networks where number of nodes are quite large, and commonality is observed which is the concern of this the thesis, the tool should work with multiple servers, central and non-central servers. This will be carried out by categorizing or forming group of nodes by their functionality and similarity in the secondary or non-central tool servers to then define each group as a node to the central tool server.

Therefore, nodes will make groups in the non-central (Secondary servers), these groups will then be defined as a node to the central server, similarly to how distributed monitoring works. This process will minimize the load on the central server as the number of groups will be much lower than checking every node separately when issues occur.

With respect to the services, Services on a certain node of the central server are defined as services if and only if that similar service is defined on at least one of nodes of the assigned group in the secondary server. In the secondary server if there is a service that has a different status on a node, the service with the worst status seen will be sent to the central server with a tag of the nodes that specific service belongs to which will help identify which node in that group is experiencing the issue.

There are two conditions that must be fulfilled for the whole design to work smoothly the first one Is hosts are grouped based on functionality. A group of Routers, core switches, distribution switches, Access points, controllers ...extra. As a part of a service, we could monitor up time, BGP peering, Virtual interfaces, physical interfaces, CPU, bandwidth, power supply unit status, temperature, fan status and more. The design has three node and five service statuses.

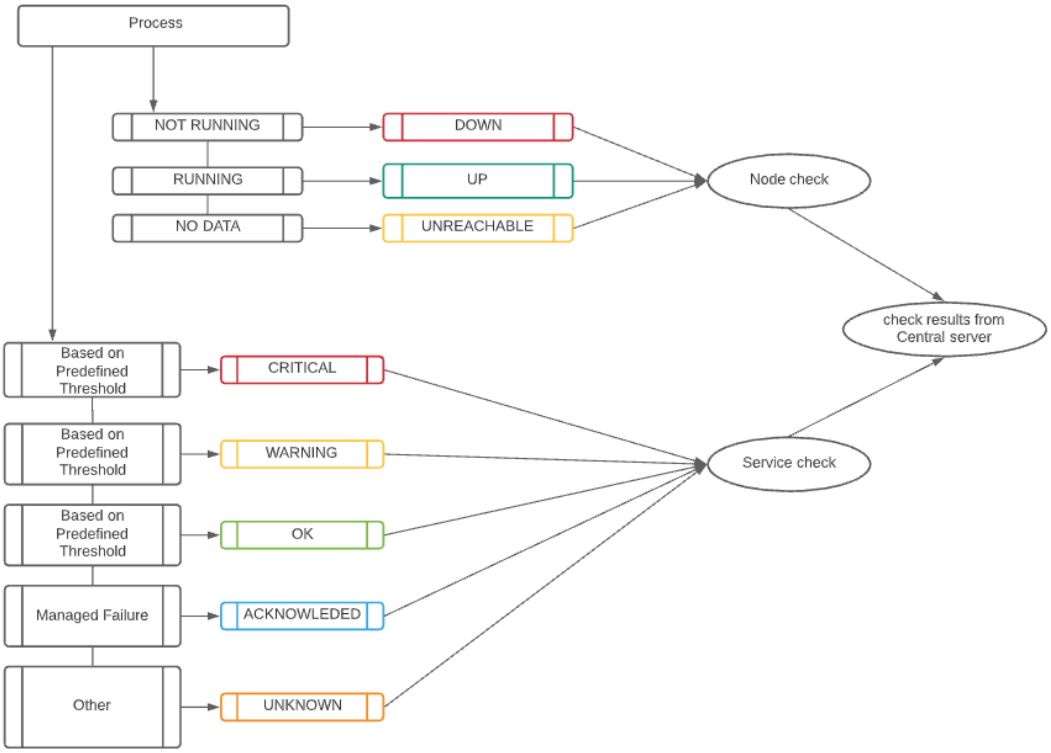


Figure 17:Node and service status labels (Author, 2021).

The statuses for service checks depends on what threshold is defined for that specific service, for instance a certain node may need its fan to operate under 60 degrees to be tabled as okay and this may not be true for other nodes depending on locations, product type, specifications and more, hence its subjective. The Unique status this design included for services is the ‘AKNOWLEDGED’ status. This status is always reserved for service issues that occurred on the non-central servers and are already acknowledged or managed.

The status labels for nodes on the other hand are common for any node under monitoring, a node that is running and stable is labeled as UP, and status shows DOWN when not running. The third status appears when there is not enough information present for the tool to label the nodes as either “UP” or “DOWN”. As per the figure 14 over all check results will have five status labels. Table 4 shows the details as below.

Table 4:Monitoring Status return values (Author, 2021).

No	Status	Description
1	OK	Nodes are performing as expected
2	WARNING	There is an event that needs attention, appears when warning limit is reached but below the defined critical limit.
3	CRITICAL	Node(s) and service(s) are not performing as expected
4	ACKNOWLEDGED	Issue is already managed or acknowledged.
5	UNKNOWN	Status is unknown. Status cannot be labeled due to lack of information.

These status reports and notifications are sent from the central tool server. When an issue occurs in the monitored infrastructure, the description of the service with issue, the node ID and its tag on the central server will be sent in an order of decreasing impact. As an example, if a BGP peer drops on 10 nodes and there were also other 20 nodes showing warning and 5 nodes with scheduled outages. The notification would look like this.

Notification subject: Critical BGP alert -Peer x.x.x.x has dropped.

Notification Body:

Critical (10) node [05,110-112,134-136,120-121,150-151]

Warning (20) node [05,110-112,133-136,120-128,150-154]] {temp sensor> 80% (Sensor Threshold Value= 2500 Sensor Value=2900)}

Acknowledged (5): node [014-016]

The numbers in the box brackets represents which node tags are involved in the issues and the numbers in the round brackets represent the number of nodes affected. This logic lowers down the number of notifications that must be sent, From the illustration there should have been 35 separate notifications sent, but this logic lowers it down to just 3. In that way it would be effortless to easily identify critical issues from non-critical briefly.

Moreover, sending out numerous alarms at once to the server to handle will make it very strenuous to identify critical issues and restore them as fast as possible. In addition to that, sending as many notifications as the monitored nodes with issues that needs attention via SMS or emails will be incompetent inefficient and hard to manage.

4.2.2 Proposed Architecture

The proposed architecture is designed in a way to provide the basic and crucial scalability for the sake of large-scale infrastructures. This is accomplished by focusing on the methodology used to collect monitoring information. As discussed in brief the design should give the flexibility of choosing the methods to collect information from monitored nodes, for this purpose the design uses a plugin-based architecture by using the concepts of Cacti, Nagios and Grafana as a groundwork reference.

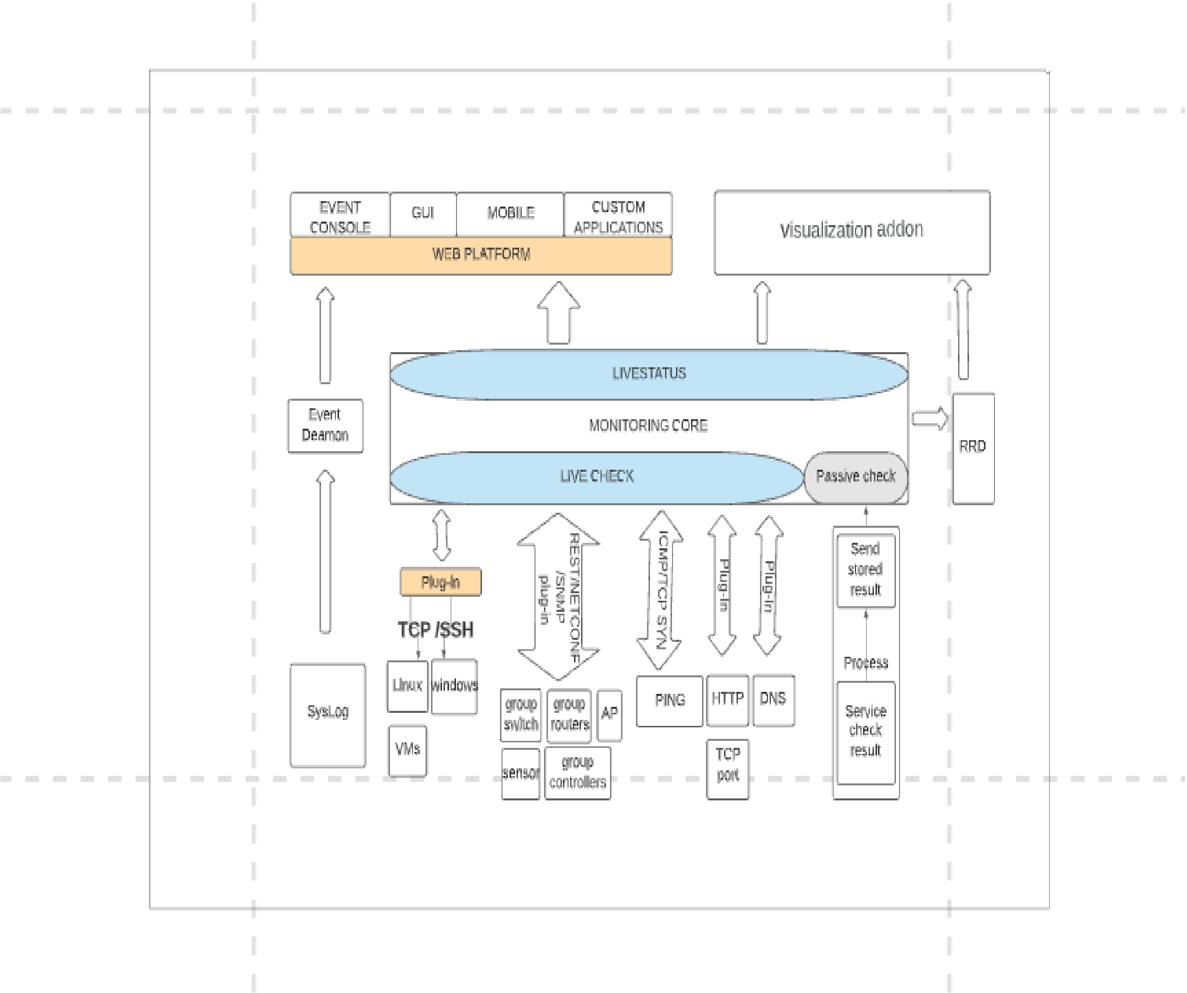


Figure 18: Proposed architecture (Author, 2021).

Generally, the design provides a twofold model where the monitoring data is pushed or pulled (based on the plugin type or methodology) towards the monitoring core or management and then passed to a platform where the status of the monitored infrastructure can be seen and managed. The major components of the proposed architecture will be explained as below.

a) Monitoring Core:

The core should act as an engine where all the monitoring information goes in to, it does not have or include an internal method like most tools to monitor infrastructures, instead it depends on external programs to provide monitoring. The core should execute these external programs to examine the status of the infrastructure under monitoring.

This can be beneficial as it provides flexibility to decide the type and form of methodology to be used to collect monitoring information. The core should also be able to receive monitoring information for passive checks, in the case of live checks the core is responsible to trigger the evaluation that it receives from the external programs.

However, for passive checks it waits until stored monitoring information gets to it. The monitoring core should process the monitoring information that it receives from these external programs and take the appropriate action as from figure 15, send out this processed information into the different web platforms. The core should also support different ways to be configured to send out notifications when needed.

b) Plug-Ins:

Plug-Ins are any programs or executable scripts that can communicate with nodes under monitoring for the purpose of collecting monitoring data. These plug-ins should be able to be developed in any language possible. Whether one decides to monitor network infrastructure via SNMP or RestConf/NetConf it should be possible if these plugins are developed, and the monitored nodes can support data models in the case of RestConf/Netconf where monitoring and configuration data encoding is carried out in text format using either Java script object notation or extensible markup language.

when in need to monitor servers and virtual machines the design recommends developing a separate plug-in. SNMP traps and syslog messages are integrated in to monitoring thought the event consol.

c) **Web platform**:

The collected monitoring information gets pushed into a web platform of one’s choice, GUI, mobile or custom applications where the status of monitored system can be visualized. What this architecture adds to the core logic besides is a service-oriented notification mechanism which the core logic by itself does not have.

d) **RRD (Round Robinson Database)**

RRD is an opensource industry standard data logging and graphing system. By the help of developed plug-ins generated monitoring data can be stored into RRD tool which can be displayed with the help of integrated display tool.

4.2.3 UML design diagrams

Aside from the detailed architecture, the system designing is carried out using the unified modelling language. These diagrams are used to show the static and dynamic aspects of the design of the tool. A class diagram is prepared to reveal the overall arrangement and relationships, A use case diagram to help design the system from the perspective of the end user representing the most important functional requirements of the design and summarize the details of the system tool design and a related sequence diagram to emphasize and show interactions among objects in time sequence. A state diagram is also prepared to show the general behavior of the system representing dynamic diagrams.

a) **Class diagram**

Figure 16 below shows a class diagram that shows classes and their basic relationship. The central logic of the design of the monitoring system is the core. class core shows the most important operations that the core should perform. Its most important duty is to store, process and evaluate measurement of monitoring data before any monitoring data is sent into any web platform. Every Agent sends the monitoring information it has to the core; therefore, the core is also responsible to assign the necessary resource to the agents. Moreover, it is required to perform plugin administration (add, or remove plugins).

Sending of monitoring alerts or events are also operations the core should carry out. The overall system is designed in a plugin-based architecture therefore principally the agent is needed to operate all plug-ins, control them, and defines resource quota or allocation for all the pug-ins. It is responsible to send the monitoring data that is determined by Plugin to the core. Plugin act as a communication mediator in between the managed nodes and the monitoring system. Class plugin performs operations like determining measurement data at a fixed time interval, mapping of the measured monitoring data and parsing it to be sent to the core via the agent. There is a one-to-one relationship between a plugin and the managed node or service. Class managed node represents the monitored devices in general.

Class monitoring data is an association class that represents the overall data that the plugin requests from the managed devices, as explained in detail under the architecture of the design monitoring is performed separately for nodes and services, they provide what is shown in detail under the association class monitoring data.

Class metric is a class where the value of resource under monitoring is defined. It represents the information about a specific service quality or grade. For instance, a certain bandwidth of a network card. Class service represents the different services that run on nodes under monitoring whereas class node represents the managed devices as discussed. Status is the state of the nodes or services or both.

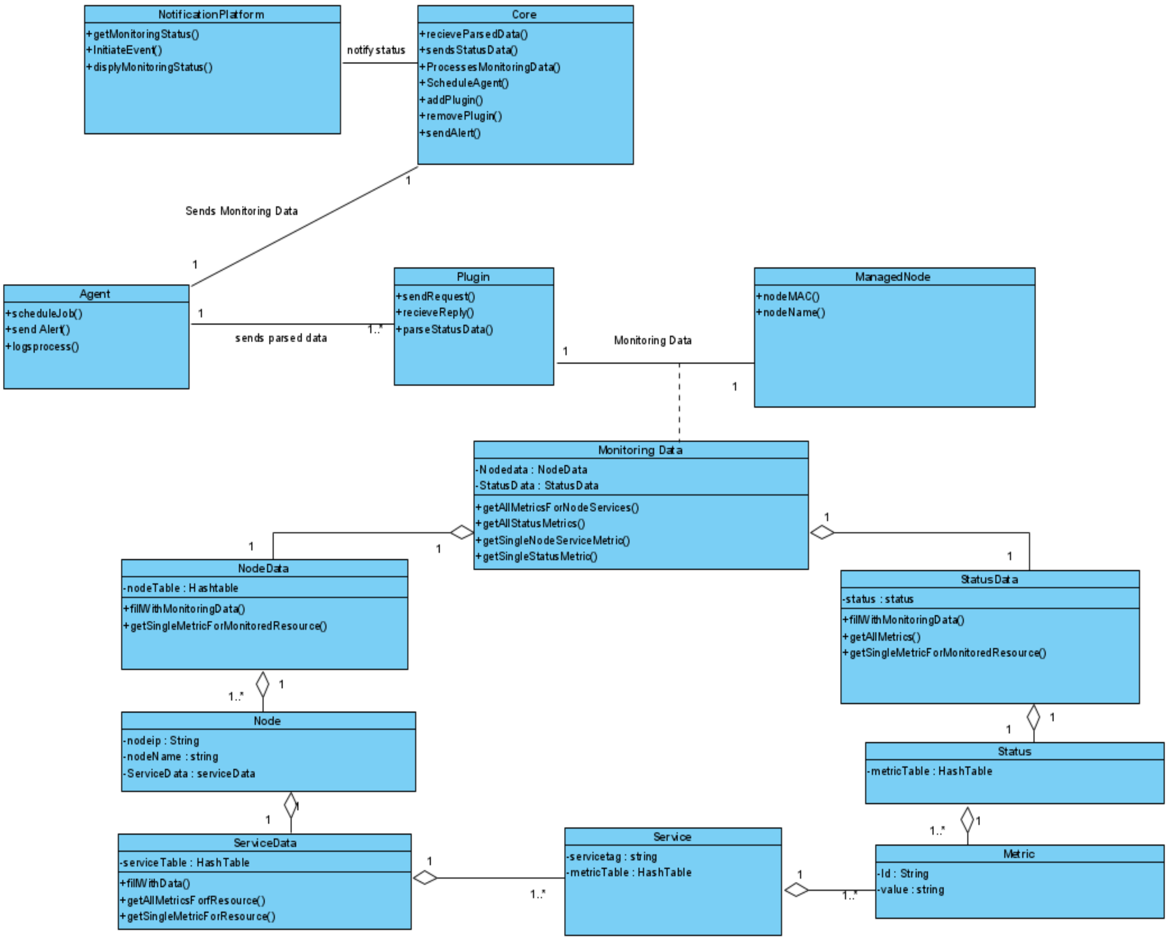


Figure 19:Class Diagram of Monitoring tool design (Author, 2021).

b) Sequence diagram

As shown in figure 17 a general sequence diagram for the monitoring design. Necessary configurations for measurement schedule are performed on the core, based on that configuration the request for measurement of data investigation goes through the plugin to the managed node.

The diagram shows the schematic communication between the core, agent and plugins which acts as an interface between the managed node and the monitoring system. Given that there could be differences in structures of monitoring data between the managed nodes and the core, mapping is necessary for the monitoring data that comes from the node which will be carried out in the plugin. parsing of these data is as well conducted in the plugin.

The received phrased measurement data is then brought back to the core. As a scenario, consider a router under monitoring for its uptime or reachability, as per choice either SNMP or RestConf plugins can be used to act in between the router and the monitoring system in general (Core and agent), these plugins will investigate, perform mapping and phrasing of the data for reachability of the router which will then be sent to the core via the agent. This process loops for as long as the router is under monitoring for its reachability.

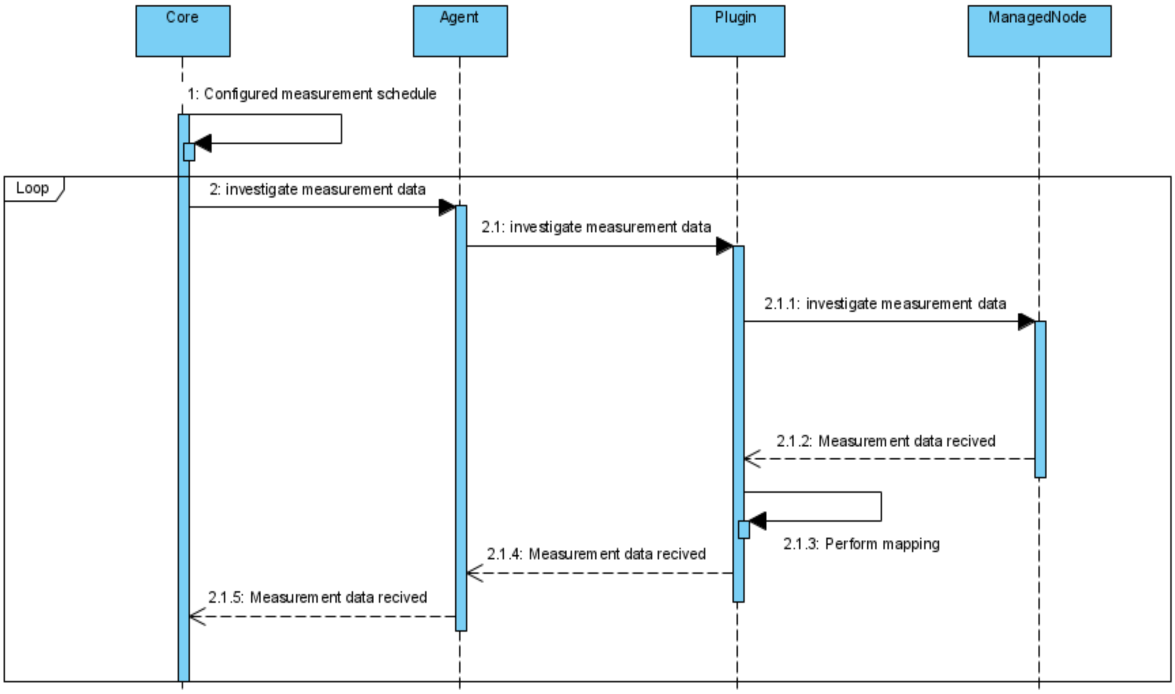

Core
Agent
Plugin
ManagedNode
1: Configured measurement schedule
Loop
2: investigate measurement data
2.1: investigate measurement data
2.1.1: investigate measurement data
2.1.2: Measurement data recived
2.1.3: Perform mapping
2.1.4: Measurement data recived
2.1.5: Measurement data recived

Figure 20:Sequence diagram of Monitoring tool design (Author, 2021).

c) Use case Diagram.

A system uses case diagram under figure 18 is prepared to demonstrate the different ways actors interact. The figure tends to show a high-level overview the monitoring system. It uses a general scenario to show the goal of the design and the scope of the system without getting into a lot of detail. A user or developer can execute scripts or plugins that can be used for monitoring different nodes, perform planned maintenance and receive events or alerts when there is an issue via a notification platform.

This user does not necessarily have to be the person who is responsible to develop the plugins or executable scripts for monitoring, the user can also be anyone who is working on the monitoring tool while, monitoring a network, performing tool maintenance, or scheduling outages. The agent can control plugins and communicate with them to send and receive node and service data. As explained in detail Notification platforms can be of different type and their use case can be different based on the type of the chosen platform. It is believed that for better visibility a web platform for GUI is mostly used to observe monitored infrastructure while the alerting mechanism might be different and could be received via an email or SMS or a call when needed.

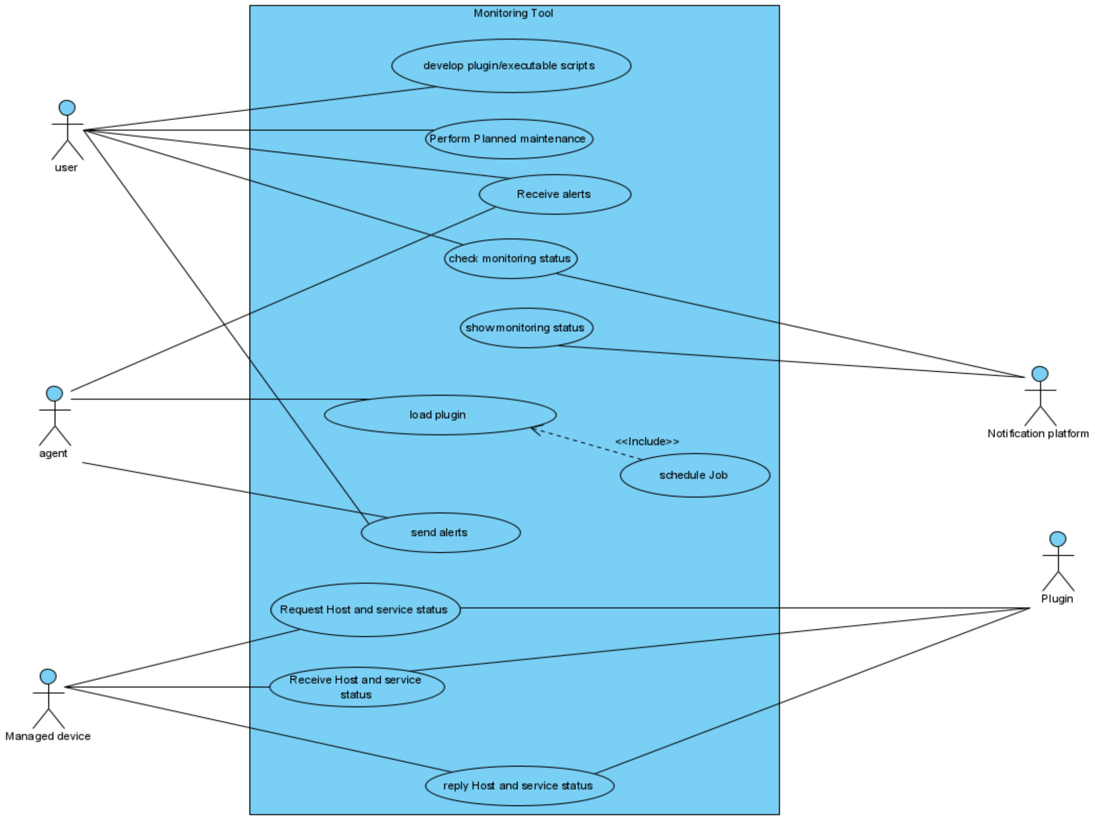


Figure 21:Use case diagram for monitoring tool design (Author, 2021).

4.2.4 Logical Data Flow Diagrams

Three levels of logical data flow diagram are shown below to graphically represent the flow of data in the monitoring system. It conveys the major processes that are associated with the system to carry and transfer data.

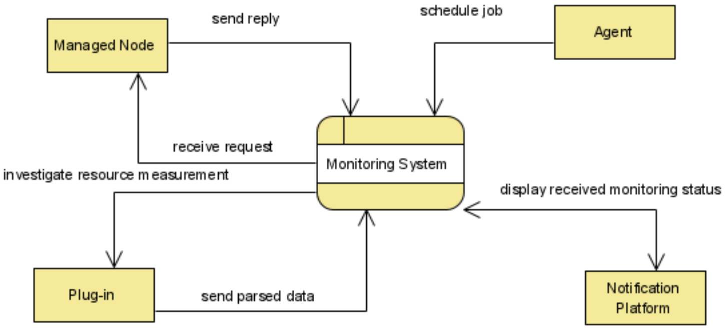


Figure 22: Context level diagram for monitoring tool design (Author, 2021).

The above figure 19 represents the level 0 data flow diagram or a context level diagram. it has one process called the monitoring system which will then be split into other major processes to give a much considerable detail. the entities as shown from figure 19 are the plug-in, managed node, agent, and the notification platform. the figure represents the complete system where monitoring data is requested and received by plugins from managed nodes to get to the core monitoring system via the agents.

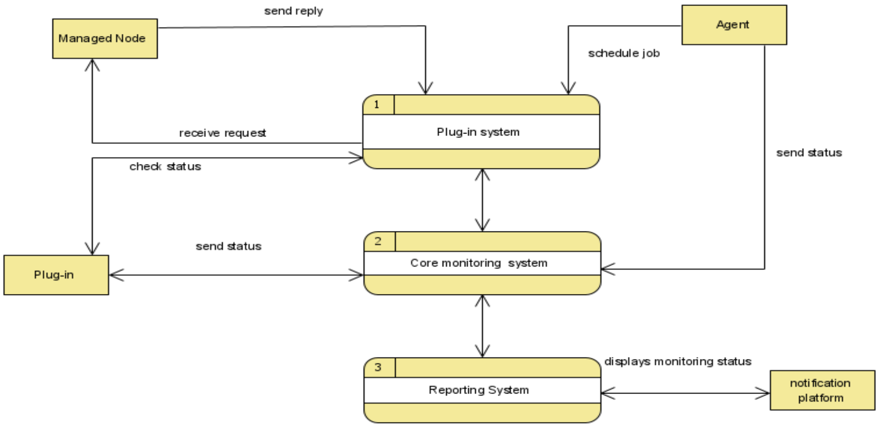


Figure 23:Level 1 DFD for monitoring tool design (Author, 2021).

A first level data flow diagram is shown under 20 the major process from the context level DFD has been split into the plugin system, the monitoring management system and the reporting system. The plug-in system as briefly explained on the thesis has the responsibility to collect the monitoring data from the managed nodes to be sent into the core monitoring system. These plugins can be a piece of software or executable scripts that are developed to work specifically for different services that are needed to be monitored in the network infrastructure. The core system is the major component of the system it stores, process and evaluate measurement of monitoring data before any it is sent into any notification platform. This data is received from the developed plugins via the agents. The reporting system then displays the gathered monitoring information. These systems will have defined procedure, process, and facilities to generate different reports of the collected network monitoring information.

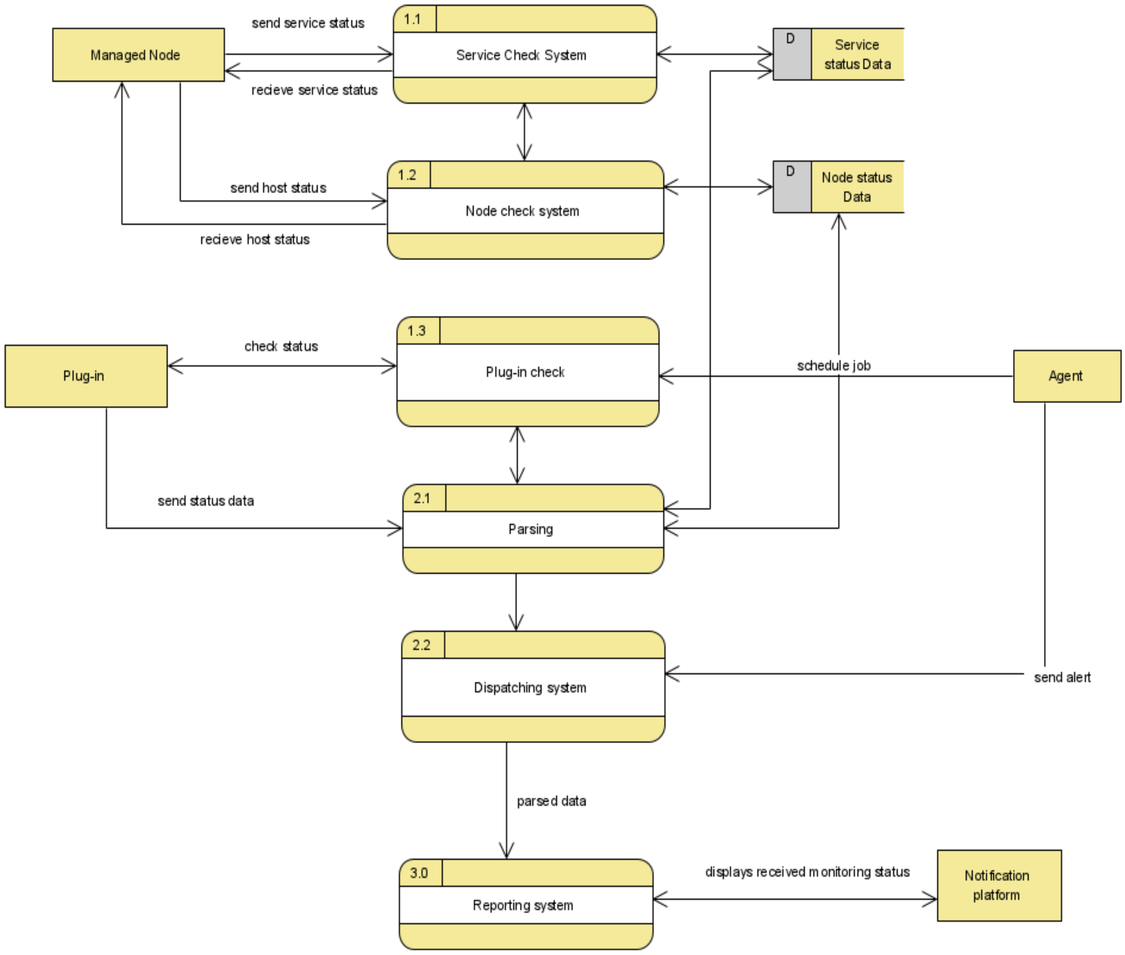


Figure 24:Level 2 DFD for monitoring tool design (Author, 2021).

Process from first level DFD are split into more detailed processes to draw the above second level data flow diagram (Figure 21). Since the system sees services and nodes separately while monitoring, service check and node check are carried out with priority given to the service check and their status are saved for plugins to parse them as explained under figure12.The service data and node data will be gathered based on the configured schedules which are collected by specific plug-ins which then needs to be phrased , parser takes the node and service data as in input in the form of sequence and produces outputs to send them to the core monitoring system via the agents which can then be dispatched to a notification platform , This can a GUI , SMS or email as a notification method .This methods are subjective and could be used based on the interest of the individual or organizations that implement the design.

5 Results and Discussion

The major component or process of monitoring networks is the methodology used to collect or gather monitoring data. A certain methodology that is used to monitor a quite simpler network might not be the best way to be used in a complex mission critical network. Consequently, based on the methodology it applies to collect monitoring data, a monitoring tool that is well suited for a simple topology may not be suitable for monitoring a large-scale network or perform as effective as it does for a less complex network. Hence an emphasis was given on what methodology or process to use while proposing a well-designed network monitoring tool (Approach)for large scale complex networks that can resolve some of the limitations or problems that could arise by using a protocol-based approach or SNMP. To achieve this, an evaluation was made on the protocol.

This evaluation was conducted on a real complex large scale production network where live traffic was running to make sure that evaluation criteria and performance indicators are well tested with a goal of identifying the protocol's limitations from two different perspectives. One of the perspectives is to evaluate the protocol's trait of intensively using resources on a network. In other words, evaluating SNMP if it affects the network it monitors by intensively using the network's resources.

The other and second perspective was to evaluate the protocol from a point of view of performance with regards to how fast it reports issues or delay. Two performance indicators or resources were chosen to evaluate the protocol from the perspective of its effect on the network it monitors, CPU and Bandwidth utilization on eight different devices (4 routers and 4 switches) taking observations of ten samples for each device under evaluation. The results of measurement on CPU utilization for SNMP polls and traps showed higher values. As it can be seen from table 1, CPU polls for routers had up to 0.79 and 0.48 on switches. Whereas Traps showed 0.77 and 0.62 for routers and switches, respectively. The measurement results shows that this whole operation and process consumes up to an average spike of 0.8. According to some vendors CPU usage p for SNMP are considered to be a low priority process. Nevertheless, compared to CPU utilization standards of different vendors [0.40-0.70] the results show moderately intensive. Measurement results also showed that relatively the routers under study showed that they suffered from high CPU spikes. These routers have overly large routing table being polled every time they try to learn routes which is the case in most large-scale complex networks. A trend was observed from the measurement results that the routers with the largest tables has shown higher CPU on the SNMP engine, implying that the larger the table the more intensive the engine CPU usage gets. Having CPU spikes can slow down telnet /SSH response which was observed on one of the four routers under evaluation. Other symptoms of intensive utilization of CPU resource by SNMP could be a slow ping response, routers struggling to send updates to their neighbors and a slow response on console which were not seen on the evaluation but are common symptoms and could be seen on a large, monitored network.

Unlike routers, Switches on the other hand did not show any symptoms even though the measurement results were moderately intensive. One trend that was observed on the switches was switches with higher number of VLAN IDs showed a relatively higher utilization in polls results. The results for band width evaluation for both routers and switches were low. Having higher band width utilization could cause packets getting dropped, high buffer failures, latency, slow performance and more which were not observed at the time of evaluation. Results imply the resource utilization is low moreover no symptoms of high band width utilization were observed.

The overall results of evaluation with the perspective of observing SNMP’s usage of resource in a monitored network and whether it affects the network it monitors or not imply that the protocol does not affect the network it monitors in the worst possible way despite to the slowness seen on SSH response for a router. There are threats to this evaluation like considering omitting other factors that can cause similar symptoms or high band width utilization to what the devices under evaluation could have shown, therefore, to blame it all on SNMP alone would be unfair, unreal and would need in depth troubleshooting. However, as for the evaluations conducted on this thesis bandwidth resource utilization is low moreover no symptoms of high band width utilization were observed. As it can be seen from table 2, results for SNMP reporting time and delay were significant. one of the samples show up 12 minutes late which could result in high business impacts. A false alarm was also observed from the eight conducted evaluations.

As per the results of evaluations conducted on this thesis, using a protocol-based approach to monitor large scale networks did not show its cons in the most convincing way. This could be due to the number of nodes, the capacity of network circuits and the amount of monitored service on the network. Yet the way SNMP operates can have some limitations on a complex network.

When working with complex and mission critical networks the effect of Pushing lots of data and bandwidth comes at cost. Deciding to put much of the traffic to go through firewalls, that would lead to a need to increase the SNMP polling to get data as quicky as possible to be more proactive and not reactive to the network and the last needed is experiencing packet loss and it might go to indicators like the CPU which is causing this issue. From the business stand organizations want to increase the monitoring thresholds by bringing it down from higher time value like 15 minutes down to 5 trying to get the system to give a lot more insight and reactiveness to be able to react to events but by pushing SNMP too hard the system is essentially pushing the devices too hard that might in turn cause outages.

When considering a more proactive approach than reactive alternatives like RestConf or NetConf should be used. Telemetry options can provide different methods of communication with monitored devices and flexible interfaces where encoding of the monitoring data is performed in a text format using JavaScript object notation.

These alternatives basically operate with a push approach than a poll which is how SNMP operates. Additionally, if ease of troubleshooting is a factor one is considering again these options bit SNMP as they do not force an engineer or any user to get on the CLI which is hard to operationalize and depend on syslog for looking into previous issues as an input for troubleshooting and observations.

These alternatives are on their very early age of usage and implementation but are undebatable the future of network monitoring and configuration management. However, SNMP has been in the world of monitoring for over 30 years now even though there will no longer be newer versions, it is implemented in over millions of systems and software and its simplicity and familiarity with Network engineers is what makes it stay with its associated problems and limitations. Moreover, results of evaluation also did not show worst results except for the protocol’s lag in reporting issues combined with other systems.

Considering these evaluation results and telemetry options as discussed, a well-designed monitoring tool (Approach) is proposed. To come up with this design that has taken the conducted evaluation and discussed factors under consideration, the best way would be to find a way to take advantage of both methodologies in the new design, meaning the conceptual design and architecture of the tool should give the flexibility of choosing which methodology the user wants to implement in the network infrastructure. A scenario could be the skills that individuals have to work with telemetry options, or the capability of the monitored devices to support data models. The proposed architecture is plugin based which gives the flexibility the design needs to choose these methodologies. Most or all tools come with internal systems making it mandatory to stick with mostly a protocol-based approach but as discussed there are limitations with using the protocol-based approach. For instance, if one needs to monitor nodes with telemetry options the new design gives them the option, oppositely if one decides to give more value to sticking to a protocol-based approach the new design guarantees it as the option is available via a plugin. Besides, it could be too early to completely move to telemetry options as most organizations would like to keep working with pure SNMP due to its familiarity and extensivity in devices, organizations may not be ready to get rid and move away from their legacy devices that does not support any data models to completely move to telemetry options.

The proposed architecture is explained in detail with its working logic. Aside from the detailed architecture the system designing is carried out using the unified modelling language. UML design diagrams where prepared to show the static and dynamic aspects of the design. Three levels of logical dataflow diagram were also prepared to show major processes that are associated with the system to operate and transfer data.

6 Conclusion

This thesis proposes a design of a monitoring tool (Approach) for highly complex or large-scale network systems that can resolve some of the limitations or problems that could arise by using a protocol-based approach or simple network management protocol. One of the most important characteristics of a well-designed monitoring tool is the methodology it uses to collect monitoring data. Hence, the thesis took a step by conducting an evaluation on SNMPv3. The evaluation was conducted on a real complex large scale production network where live traffic was running to make sure that evaluation criteria and performance indicators are well tested. The perspectives of the evaluations performed were to investigate if the protocol intensively uses the resource it monitors resulting in affecting the network under monitoring and examining the performance of the protocol by investigating its reporting time or delay when an issue or event occurs on the managed network. The results from these evaluations did not completely show that using a protocol-based approach on a large-scale network affects the monitored network in the most convincing way, however a significant amount of delay in reporting issue was observed. The thesis also discusses telemetry options that can provide different methods of communication with monitored devices and flexible interfaces where encoding of the monitoring data is performed in a text format using JavaScript object notation in detail and analyzes all the possible factors one could consider when deciding to choose a methodology to monitor Network systems. Accordingly, it proposes an architectural design with detailed working logic. Besides to the Proposed architecture a more general design of the system was carried out using the unified modelling language UML design diagrams. Major processes that are associated with the system to carry and transfer data were shown by using three levels of logical data flow diagrams.

Inconclusion, a more interesting work can be done to test the performance of SNMP on enterprise network for other performance indicators that are not included in this thesis or other perspectives that this thesis omitted to present due to scope limitations and other constraints. A step-by-step implementation of the proposed design could also make a great contribution to a smooth transition from the legacy SNMP to telemetry options for network monitoring and management that the IT industry is going through since 2018.

7 References

AL SHIDHANI, Ali, AL MAAWALI, Khalil, AL ABRI, Dawood and BOURDOUCEN, Hadj, 2016. A Comparative Analysis of Open Source Network Monitoring Tools. *International Journal of Open Source Software and Processes (IJOSSP)*. 2016. Vol. 7, no. 2, p. 1–19.

AMIRTHALINGAM, K and MOORHEAD, Robert J, 1995. SNMP-an overview of its merits and demerits. In: *Proceedings of the Twenty-Seventh Southeastern Symposium on System Theory*. 1995. p. 180–183.

BHARADWAJ, Krishna, FLORES, Samuel, RODRIGUEZ, Joshua, LONG, Lance and MARAI, G. Elisabeta, 2016. Developing a scalable SNMP monitor. In: *Proceedings - 2016 IEEE 30th International Parallel and Distributed Processing Symposium, IPDPS 2016*. 2016. p. 1043–1047. ISBN 9781509021406.

BIBBS, Eddie, MATT, Brandon and TANG, Xin, 2006. Comparison of SNMP Versions 1, 2 and 3. *Extra do el*. 2006. Vol. 17.

BROCADE COMMUNICATIONS SYSTEMS, Incorporated, 2015. Network OS MIB Reference. *Supporting Network OS 6.0.1*. 2015. Vol. I, p. 5–6.

CHATZIMISIOS, Periklis, 2004. Security issues and vuluerabilities of the snmp protocol. In: *(ICEEE). 1st International Conference on Electrical and Electronics Engineering, 2004.* 2004. p. 74–77.

DELSING, Jerker, ELIASSON, Jens and LEIJON, Viktor, 2010. Latency and packet loss of an interferred 802.15. 4 channel in an industrial environment. In: *2010 Fourth International Conference on Sensor Technologies and Applications*. 2010. p. 33–38.

ENTERPRISES, Nagios, 2017. *Nagios*. 2017.

GAILLARD, Guillaume, BARTHEL, Dominique, THEOLEYRE, Fabrice and VALOIS, Fabrice, 2016. Monitoring KPIs in synchronized FTDMA multi-hop wireless networks. In: *2016 Wireless Days (WD)*. 2016. p. 1–6.

GAMALIELSSON, Jonas, LUNDELL, Björn and LINGS, Brian, 2010. The Nagios community: An extended quantitative analysis. In: *IFIP International Conference on Open Source Systems*. 2010. p. 85–96.

GEHLBACH, Jeff, 2015. Managing Networks in a Software-Defined Future. *Southern California Linux Expo*. 2015. P. 9.

GORANSSON, Paul, BLACK, Chuck and CULVER, Timothy, 2016. *Software defined networks: a comprehensive approach*. Morgan Kaufmann.

HALL, James, 2003. *Multi-layer network monitoring and analysis*.

HAMID, Ahmad Kamil Abdul, KAWAHARA, Yoshihiro and ASAMI, Tohru, 2010. Web cache design for efficient SNMP monitoring towards realizing globalization of network management. In: *2010 INFOCOM IEEE Conference on Computer Communications Workshops*. 2010. p. 1–5.

HARE, Chris, 2011. *Simple Network Management Protocol (SNMP).* 2011.

HE-WEN, YANG, 2015. Network Management System Based on OpenNMS. *Computer and Modernization*. 2015. No. 10, p. 88.

HELD, Gilbert, 2002. *Ethernet Networks: Design, Implementation, Operation, 1/ 2Management*. John Wiley \& Sons.

JACKSON, Alden W, STERBENZ, James P G, CONDELL, Matthew N and HAIN, Regina Rosales, 2002. Active network monitoring and control: the SENCOMM architecture and implementation. In: *Proceedings DARPA Active Networks Conference and Exposition*. 2002. p. 379–393.

JIN, Xin, HA, Tal Soo and SMITH, Dean P, 2008. SNMP is a signaling component required for pheromone sensitivity in Drosophila. *Proceedings of the National Academy of Sciences*. 2008. Vol. 105, no. 31, p. 10996–11001.

JUAN, W U, 2016. Implementation of SNMP Protocol ups Data Acquisition Based on ICS SNMP. *Journal of Anhui Vocational College of Metallurgy and Technology*. 2016. P. 4.

KULKARNI, Sameer G, LIU, Guyue, RAMAKRISHNAN, K K, ARUMAITHURAI, Mayutan, WOOD, Timothy and FU, Xiaoming, 2018. Reinforce: Achieving efficient failure resiliency for network function virtualization based services. In: *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. 2018. p. 41–53.

LANDFELDT, Björn, SOOKAVATANA, Pipat and SENEVIRATNE, Aruna, 2000. The case for a hybrid passive/active network monitoring scheme in the wireless Internet. In: *Proceedings IEEE International Conference on Networks 2000 (ICON 2000). Networking Trends and Challenges in the New Millennium*. 2000. p. 139–143.

MAHAJAN, Aman, JOSHI, Haresh and KHAJURIA, Sahil, 2012. ICMP SNMP?: Collaborative Approach to Network Discovery and Monitoring. *Network*. 2012. Vol. 1, no. 3, p. 16.

MAURO, Douglas and SCHMIDT, Kevin, 2005. *Essential SNMP: Help for System and Network Administrators*. " O'Reilly Media, Inc."

MICHALAK, Damian, 2018. SNMP in theory - simple and proven network monitoring. [online]. 2018. [Accessed 9 March 2021]. Available from: https://www.nastykusieci.pl/snmp-teoria/

MORRIS, Stephen B, 2003. *Network management, MIBs and MPLS*. Prentice Hall Professional.

MURRAY, Peter and STALVIG, Paul, 2008. SNMP: simplified. *F5 White Paper*. 2008.

OLUPS, Rihards, 2010. *Zabbix 1.8 network monitoring*. Packt Publishing Ltd.

OLUPS, Rihards, 2016. *Zabbix Network Monitoring*. Packt Publishing Ltd.

PRESUHN, R, 2002. *RFC3418: Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*. 2002. RFC Editor.

PRESUHN, Randy, CASE, J, MCCLOGHRIE, K, ROSE, M and WALDBUSSER, S, 2002. *Management information base (MIB) for the simple network management protocol (SNMP)*.

QOS, SIMULASI D A N ANALISIS KINERJA, 2018. Simple Network Management Protocol (SNMP). . 2018.

ROSE, Marshall T and MCCLOGHRIE, Keith, 1990. *RFC1155: Structure and identification of management information for TCP/IP-based internets*. 1990. RFC Editor.

SANTOS, Paulo Roberto da Paz Ferraz, ESTEVES, Rafael Pereira and GRANVILLE, Lisandro Zambenedetti, 2015. Evaluating SNMP, NETCONF, and RESTful web services for router virtualization management. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015. p. 122–130.

SCHÖNWÄLDER, Jürgen and MARINOV, Vladislav, 2011. On the impact of security protocols on the performance of SNMP. *IEEE Transactions on Network and Service Management*. 2011. Vol. 8, no. 1, p. 52–64. DOI 10.1109/TNSM.2011.012111.00011.

SHAFFI, Abubucker Samsudeen and AL-OBAIDY, Mohanned, 2013. Managing network components using SNMP. *International Journal*. 2013. Vol. 2, no. 3, p. 1493–2305.

SHAMSI, Jawwad and BROCMEYER, Monica, [no date]. *Chapter 1 principles of network monitoring*.

SLABICKI, Mariusz and GROCHLA, Krzysztof, 2016. Performance evaluation of CoAP, SNMP and NETCONF protocols in fog computing architecture. In: *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. 2016. p. 1315–1319. ISBN 9781509002238.

STEINIGER, Stefan and BOCHER, Erwan, 2009. An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science*. 2009. Vol. 23, no. 10, p. 1345–1370.

STEINKE, Steve, 2003. *Network Tutorial: A Complete Introduction to Networks Includes Glossary of Networking Terms*. CRC Press.

SVOBODA, Jakub, GHAFIR, Ibrahim, PRENOSIL, Vaclav and OTHERS, 2015. Network monitoring approaches: An overview. *Int J Adv Comput Netw Secur*. 2015. Vol. 5, no. 2, p. 88–93.

TOM FOOTTIT, 2020. Active monitoring 101: A history of network performance data collection. *Accedian* [online]. 2020. [Accessed 9 March 2021]. Available from: active-monitoring-101-a-history-of-network-performance-data-collection/

WELZL, Michael, 2005. Network congestion control. *Proc. of Managing Internet Traffic*. 2005.