The research focuses on designing and implementing a system for managing incoming server traffic using the Simple Network Management Protocol (SNMP). The essence of the method is to provide a real-time monitoring system that empowers network managers to oversee all network activities efficiently. The key difference of the developed system lies in using the SNMP protocol to monitor and control the network without needing a third-party tool. The proposed approach addresses administrators' common network challenges, including inefficient bandwidth utilization and complex network resource monitoring. Upon implementation, the system gathered essential network equipment information from the Management Information Base (MIB). The research results indicate that the SNMP-based network management system prevents network devices from reaching their allotted bandwidth limits. The system notably reduces CPU utilization from 28 % to 8 % and decreases bandwidth usage from 1.7 MiB/s to 260 bytes/s. Additionally, the system's alerting capability enhances the network's resilience against incoming traffic anomalies that may otherwise interrupt server operations. An example of using the proposed system to manage incoming server traffic in small networks. It has several practical uses in managing and maintaining network infrastructure in different fields. Also, it enables real-time monitoring of network devices, resource management, fault detection, and performance optimization. The system can be used in small organizations to monitor network performance, track faults, and manage network resources efficiently. In addition, SNMP-based network monitoring systems can find applications in almost any field where networked devices and infrastructure require monitoring, management, and optimization

Keywords: SNMP, Real-time monitoring and management system, MIB, Traffic server management

UDC 004
DOI: 10.15587/1729-4061.2023.286528

DESIGN AND IMPLEMENT A REAL-TIME NETWORK TRAFFIC MANAGEMENT SYSTEM USING SNMP PROTOCOL

Ahmed Hazim Alhilali
Corresponding author
Lecturer*
E-mail: ahmed.alhilali@uokufa.edu.iq
Ali Al Farawn
Lecturer*
Ahmed Yaseen Mjhool
Lecturer*
*Information Technology Research and Development Center
University of Kufa
Kufa str., Kufa, Najaf Governorate, Iraq, 54001

Received date 04.08.2023
Accepted date 14.10.2023
Published date 30.10.2023

How to Cite: *Alhilali, A. H., Al Farawn, A., Mjhool, A. Y. (2023). Design and implement a real-time network traffic management system using SNMP protocol. Eastern-European Journal of Enterprise Technologies, 5 (9 (125)), 35–44. doi: https://doi.org/10.15587/1729-4061.2023.286528*

1. Introduction

As computer netorks and communication continue to grow, they have become indispensable for everyday communication. The need for computer network management emerged as network technologies rapidly evolved, coinciding with a notable reduction in the cost of computing resources [1]. While current network management software primarily concentrates on network links and equipment, it is crucial to recognize the importance of servers as the backbone of network services. Enterprises, educational institutions, industrial companies, and government agencies often encounter similar challenges related to their network performance [2]. The network infrastructure might face challenges arising from the growing integration of ICT, the emergence of resource-intensive information processing technologies, and the utilization of network resources in educational contexts [3]. Additionally, these challenges encompassed issues such as slow data transfer rates and inefficient use of available bandwidth [4]. Failure to appropriately resolve this problem can result in unnecessary expenses, as bandwidth is continuously upgraded without considering its efficiency of utilization. Consequently, there is a pressing need for an effective approach and concept to enhance network performance and optimize resource usage. However, network administrators encounter challenges when it comes to monitoring network resources within an organization. The efficient monitoring and management of network devices have emerged as challenges when establishing internal networks for large and medium-sized enterprises. The performance status of network equipment, servers, and applications directly affects the overall business quality. In addition, network administrators needed help with the issue of slow fault detection in network devices like routers, switches, cables, and other components. The delay in identifying the damaged or malfunctioning devices can significantly affect network performance [5].

The utilization of the SNMP approach in network monitoring systems simplifies the task of network administrators in centrally managing the network [6, 7]. This approach enables real-time monitoring of computer networks, ensuring their continuous security, stability, and availability to meet users' demands [8]. The current state of information society requires the highest level of protection for information resources and critical data, placing significant demands on software and hardware within systems that provide

reliable and high-quality IT services [9]. IoT management implementation has been categorized into four trends, distinguished by the management mechanism or standard employed: SNMP, TMN, WBEM, and PBM. Among these, SNMP stood out as the most favourable standard for IoT management implementation, owing to its widespread adoption by manufacturers [6]. Most device manufacturers have incorporated a network interface into their products, enabling the management of these devices through SNMP when Windows or other operating systems are installed [10]. However, many devices offer SNMP support even without an operating system installed. Utilizing SNMP has become a widely adopted approach for monitoring and controlling the status of network devices, as well as analyzing data flows. Therefore, real-time monitoring system research and development are crucial instruments for preserving the performance, stability, and security of diverse systems and environments. They help organizations respond swiftly to problems, make choices based on current information, and guarantee a safe and secure user experience.

2. Literature review and problem statement

Monitoring and detecting systems using the SNMP protocol, which employs snmpwalk command procedures, provide a means to observe and gather network-related data. Many studies used SNMP only to monitor the computer network's activities. However, the third version of the SNMP provided the ability to control the network equipment through the set command. The paper [11] presented a layered multi-agent system that consists of three agent types to monitor the network of Campus Infrastructure and maintain the data security. Where the compelling features provided by instant messengers have encouraged the students to divulge their personal information online. Consequently, the security and privacy of the data gathered by these applications have emerged as a concern, especially in light of sensitive and confidential information [12]. One of these agents is the collector agent, responsible for gathering information from network equipment using Management Information Base (MIB) values. The consolidator agent processes the collected data into a usable format, utilizing an input table of information from network devices. Finally, the application agent represents this processed data as a web service, specifically a heat map. The results showed that the suggested architecture demonstrate reduction in time, which involves a large number of agents traversing the network within distinct zones. However, it is important to note that the SNMP's behavior in the network may potentially extend the time due to increased network traffic. This heightened network activity could lead to SNMP errors that affect the measurements and affect a specific group of users. The authors in [13] employed Cacti as a monitoring tool to oversee and troubleshoot network issues across all devices within the network. They utilized SNMP to gather comprehensive information on the activities of the network devices. By using Cacti, network administrators gain the ability to analyze device usage and identify users accessing specific devices. They also received analysis results highlighting problematic server conditions, such as server memory reaching total capacity or server shutdown occurrences. However, when the system reaches peak usage, notifications should be sent in the form of emails so that network administrators can get information more quickly when not on site. While this paper [14] presented a solution that utilizes the SNMP Proxy Agent to gather IoT information from IoT communication and the local gateway's IoT log file, the agent transformed the collected data into MIB objects and then supplied them to the SNMP monitoring system. Consequently, details regarding IoT devices are seamlessly integrated into the network management system deployed locally. The primary benefit of this approach is a cohesive perspective on interconnected IoT and non-IoT devices, irrespective of the communication protocol they use. Nevertheless, the proposed technique does not focus on controlling the IoT devices using the SNMP SET command; instead, it aims to collect data from these devices for analytical purposes. Work by [15] presented a designed and examined spacecraft network monitoring system that relies on SNMP. The system incorporates a distributed terminal, enabling real-time monitoring of network devices. Testers have the capability to observe network devices, collect data, and store operation logs directly on the terminal. This enhances the efficiency of spacecraft network testing. Accumulated practical experiences have demonstrated the versatility and dependability of the network monitoring system when applied in spacecraft integration testing. Furthermore, the system's effectiveness has been substantiated through testing within the context of managing spacecraft programs. However, the method displays only monitoring information, such as real-time values and statuses of any managed objects, but there is no management portal. In addition, this study [16] employed QoS and SNMP systems to oversee and regulate the Internet network infrastructure utilized by various agencies connected to the DNIC server via the Network Operation System (NOS) department. The advancement of information and control systems enables their utilization for automating control procedures, encompassing intricate systems and essential infrastructure installations [17]. The SNMP-based monitoring system demonstrated that network administrators received notifications within less than 5 minutes, enabling technicians to promptly identify devices affected by faults. The findings suggested that SNMP-based system monitoring services were highly effective in providing quick notifications when dealing with issues in intricate networks. Moreover, the system facilitated centralized monitoring of all network infrastructure activities, simplifying network administrators' tasks. Nevertheless, it is essential to acknowledge that the current approach needs to incorporate combinations or hybrid methods to enhance monitoring performance and bandwidth management. Furthermore, the study suggested integrating additional third-party tools like Packet Shaper and Nagios Core to support a centralized system control and early warning system. The paper [18] aimed to find the ideal number of mobile agents needed for efficient data retrieval while minimizing routing time and network bandwidth impact. It addresses centralization issues related to the Simple Network Management Protocol-Management Information Base (SNMP-MIB) and seeks to enhance detection time. The suggested approach utilized two types of agents: link agents for network discovery and data agents for MIB data collection. The study initially employs a link agent to discover and connect nodes within the network. Subsequently, the network is partitioned based on execution time, with each partition assigned a single mobile agent for MIB retrieval, optimizing retrieval time and managing agent generation for network bandwidth. The findings demonstrate that the model outperforms alternative models when applied

in distributed networks with numerous nodes, particularly regarding agent quantity and time. Nevertheless, the proposed approach solely relies on SNMP for gathering data about network devices and employs this information to detect malicious activities and anomalies. Work by [19] aimed to create and implement the Simple Network Management Protocol (SNMP) for network monitoring directly at the Information Technology and Telecommunications Technical Implementation Unit (UPT TIK) within Riau University. The system is a broader network management system component, serving network administrators in device monitoring and control. Through system testing and analysis, it is evident that the designed monitoring system effectively monitors network devices. The system is designed to present crucial network management information, including device status (up/down) and traffic. Additionally, it features an alert mechanism, employing sound notifications when device statuses go offline. However, the primary system is used to reactivate inactive network devices, as the proposed system is considered a monitoring system only.

Therefore, the examination of relevant literature indicated that the majority of articles have utilized SNMP primarily for network monitoring purposes. However, SNMP's third version introduced the capability to manage network equipment using the set command. Employing this feature would result in a decreased need for third-party software or hardware to control network devices, thereby reducing associated costs. Let's use the SNMP protocol in our proposed system because it focused on providing real-time information about the network devices' performance and status. Therefore, it makes managing network devices easier than SIEM and IDS as they focus more on security event monitoring or threat detection [20].

3. The aim and objectives of the study

The aim of this study is designing a real-time network traffic management system using SNMP protocol. The system will be deployed on a server to effectively manage the incoming traffic via SNMP architecture.

To achieve the aim, the following objectives are accomplished:

– to develop a system that monitor and control the server incoming traffic through SNMP agent ports on switch;

– to integrate an algorithm into the system to detect excessive traffic on a specific port, temporarily disabling it and then automatically reactivating it after a set period;

– to evaluate the system's effectiveness in real network scenarios and assess its ability to manage and mitigate excessive incoming traffic on network ports.

4. Materials and methods

4. 1. Simple Network Management Protocol (SNMP)

The Internet Engineering Task Force (IETF) introduced SNMP in 1987 to create a universal management protocol to accommodate the growing number of network devices. The SNMP is an established Internet Standard protocol utilized to gather and structure data about managed devices on IP networks. It enables the modification of this information to influence the behavior of the devices [21]. Over time, SNMP has undergone three versions v1, v2 (including v2c), and v3. The newer version is designed to be backward compatible with the older versions, ensuring interoperability and a smooth transition between them [15]. The SNMP command is actively utilized to enable communication between the manager and the SNMP agent. SNMP encompasses three crucial components: the manager, agent, and management information base [22]:

a) SNMP management model.

The SNMP architecture, as illustrated in Fig. 1, involves the SNMP manager, responsible for monitoring and controlling the network devices, and the managed device, a network node or element supporting the SNMP protocol. Examples of such devices include routers, switches, and other equipment. Finally, the SNMP Agent is a software module for network management installed on a managed device. It is responsible for gathering management information from the local machine and converting it into a format compatible with the SNMP protocol. The SNMP Agent bridges the device's internal management data and the SNMP protocol for effective communication and interaction with the SNMP Manager [23].

The SNMP Manager is used network management software to manage and monitor the network devices. The network management software relies on the SNMP Manager to periodically gather essential device information [24]. The collected information used to assess the condition of individual network equipment, specific sections of the network, or the entire network, ensuring that everything is functioning normally. Furthermore, the gathered information can be utilized to identify abnormal activities that have the potential to cause network failures [25]. The SNMP Manager consistently requests relevant details about the operational status, configurations, and performance metrics from the agents residing on the network devices.

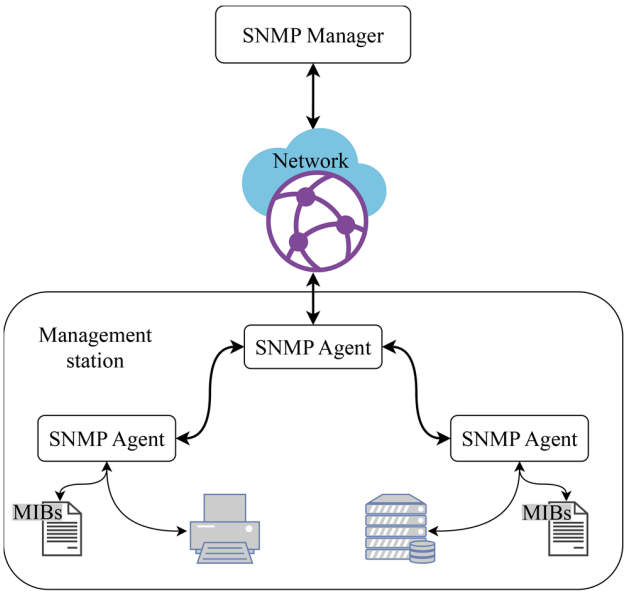

Fig. 1. Simple Network Management Protocol architecture

b) SNMP Management Information Base (MIB).

Managing network resources involves using objects to represent specific aspects of the resources. The SNMP protocol has a hierarchical structure known as the Management Information Base (MIB). The MIB serves as a database containing information about the network and the

managed devices. It consists of managed objects organized hierarchically. By reading and setting the values of these MIB objects, the management station can perform comprehensive monitoring and control functions [14]. The agents on the backbone devices maintain their own MIB to reflect the status of managed resources. The network management entity can monitor nodes by accessing the object values in the MIB, and it can exert control over the resources by modifying the values of the objects [10]. Within the MIB, there are two categories of objects: scalar and tabular. The Object Identifier (OID) structure follows a tree-like hierarchy and is represented by a series of integers separated by dots. This OID system assigns a unique ID to each object, such as devices (e. g., router, switch), interface names, interface states, and other relevant information, facilitating their instantiation and identification within the MIB [16, 26]. Fig. 2 illustrates the hierarchy and format of the MIB object identifier [27].

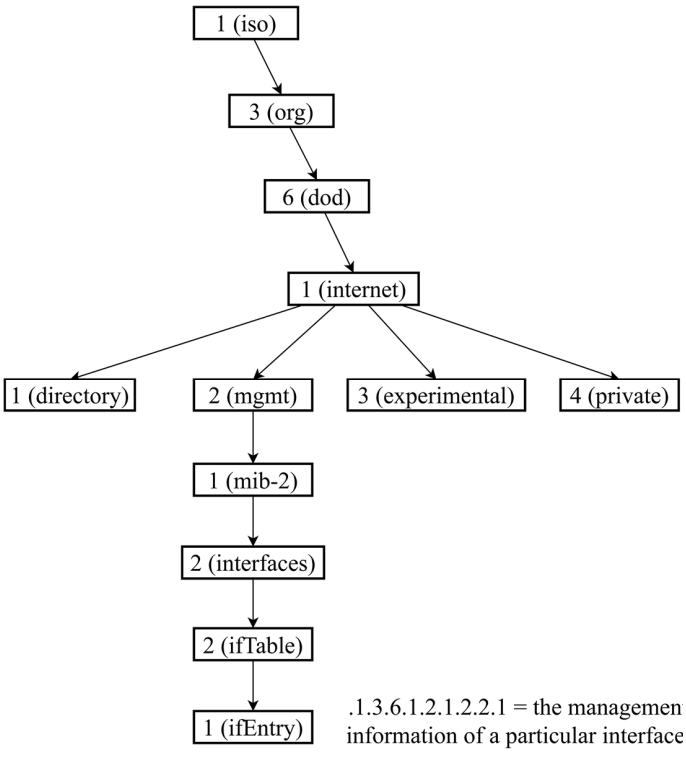

Fig. 2. Management Information Base Object Identifier Hierarchy

c) SNMP commands.

SNMP operates as an application layer protocol within the OSI five-layer network model, making it easily applicable in various network environments. SNMP's Product Data Unit (PDU) utilizes the User Datagram Protocol (UDP) in the transport layer. This ensures reliable SNMP PDUs' transportation through SNMP's built-in data services, including mechanisms like re-transportation in case of timeouts [11]link utilization, quality of service (QoS. SNMP Managers and SNMP Agents communicate with each other via SNMP messages. These messages perform tasks, including information retrieval, command execution, and notification transmission [28] IoT service gateway including convergence platform could be supported on dynamic module system that is required mounting and recognized intelligent status with the remote network protocol. These awareness concepts support the dynamic environment of the cross-platform distributed computing technology is supported by these idea as a Universal Middleware for network substitution. Distribution system commonly used in recent embedded systems include CORBA (Common Object Request Broker Architecture. SNMP messages come in a variety of formats, as follows:

– SNMP Get Request: to query detailed information about a managed device or object, the SNMP Manager sends a Get Request message to the SNMP Agent;

– SNMP Get Response: in response to a Get Request, the SNMP Agent notifies the SNMP Manager with a Get Response message. If the request cannot be performed, this message will provide either the required information or an error message;

– SNMP Set Request: the SNMP Manager uses a Set Request message to change the value of a particular item on the SNMP Agent. The agent is given instructions in this message to modify the value of the mentioned object;

– SNMP Set Response: a Set Response message, which the SNMP Agent sends in response to a Set Request, reports whether the desired value change was successful or unsuccessful;

– SNMP trap: to inform the SNMP Manager of a significant occurrence or circumstance, a SNMP Agent might send an unsolicited Trap message. Traps are used for alerting and proactive monitoring;

– these SNMP messages followed a specific structure established by the SNMP protocol. The information required for communication between the SNMP Manager and SNMP Agent is carried through their headers and payloads. The message's underlying transport technology is commonly UDP (User Datagram technology).

5. Results of research on the design and implement a real-time network traffic management system using SNMP protocol

5. 1. Description of the system design

The Management System (MS) primary purpose, which is meant to run on a server linked to a network switch, is to regulate each switch port's status according to the amount of incoming traffic. The steps listed below describe how the system operates:

– system deployment: a dedicated server is used to host the system. This server has the required access and permissions to communicate with the network switch and is connected to it through an Ethernet interface. The system's input consists of network traffic data gathered by the SNMP agent (in our case, the IP Switch), encompassing information about the status and performance of the managed network device;

– monitoring of incoming traffic: the system monitors all network traffic to the switch ports. It provides real-time measurements of the traffic volume and capacity of each port;

– traffic threshold: a precise 30 % threshold is set up in the system. The maximum percentage of the entire port capacity that incoming traffic should not exceed is represented by this threshold;

– actions for Port Management:

a) overcoming threshold: the system will issue a management order to deactivate a specific switch port if the incoming traffic on that port is above 30 % of the port's

total capacity. By taking this move, any additional traffic is essentially prevented from using that port;

b) reactivation: the system will automatically send the necessary management instructions to restart the port after a minute.

The system design begins with establishing the network and configuring its associated devices. Subsequently, the configuration of the SNMP agent and manager, the main components of the proposed system, is undertaken. Below are the comprehensive steps that explain the system design process:

a) Build and configure the network.

Let's consider a network with the following architecture (shown in the Fig. 3) in which several computers are connected to an IP switch; each of these computers is sending specific traffic to the server. The network was built using GNS3 (Graphical Network Simulator-3), a free, open-source, and dependable graphical network emulator. It empowers network engineers to construct and evaluate a simulated network model, complete with its topology and application flow while considering that the existing network is being implemented in the real world [29]. In our work, the network has a server with Linux Ubuntu OS; also, it has regular hosts (A/B/C) that will used to generate traffic. A virtual machine called traffichost1 (Fig. 3) will be used to generate rapid traffic during the system execution time in the testing process. Table 1 includes the configuration of the network devices including device name, interface, IP address, subnet mask, and default gateway.

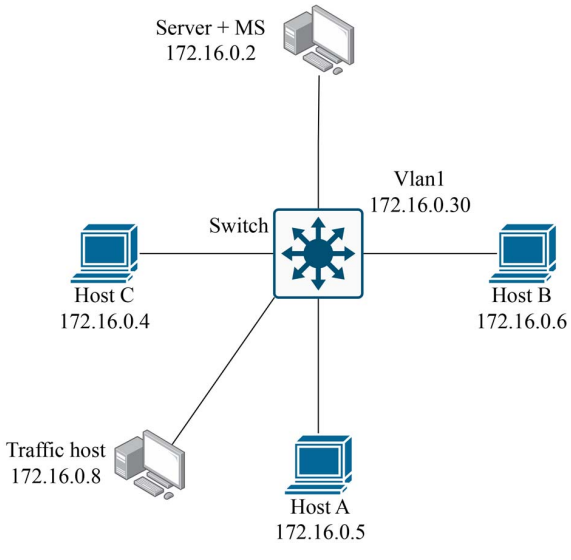

Fig. 3. The network topology

Table 1

The network devices configuration details

Device	Interface	IP Address	Subnet Mask	Default Gateway
Server+MS	Ethernet0/0	172.16.0.2	255.255.255.0	172.16.0.1
Switch	VLAN 1	172.16.0.30	172.16.0.1	Blank
HostA	Ethernet0/2	172.16.0.5	255.255.255.0	172.16.0.1
HostB	Ethernet0/1	172.16.0.6	255.255.255.0	172.16.0.1
HostC	Ethernet0/3	172.16.0.4	255.255.255.0	172.16.0.1
traffichost-1	Ethernet1/0	172.16.0.8	255.255.255.0	172.16.0.1

b) SNMP Agent – Manager Configuration.

The following steps are generally necessary to enable the SNMP Agent on a device (in our case, the IP Switch):

– access the management interface: Connect to the device through the proper administration interface, like a command-line interface (CLI) and enable the SNMP functionality;

– configure SNMP settings: including SNMP community strings (NMS), which are used for authentication and access control, SNMP server location, contact, and access list. Let's apply the commands shown in the Fig. 4. from the global configuration mode. In the first line, the SNMP community string is configured, with read-write privileges. In lines 2 and 3, the SNMP manager location and contact information are provided. In line 4, all default SNMP traps are enabled, and line 5 specifies the name of the access list. Finally, allow which hosts can get SNMP information from the Switch in line 6;

– save and apply the modifications: When the SNMP Agent activated and configured on the device, all changes should be saved. By doing this, the SNMP Agent is guaranteed to start up and be prepared to speak with the SNMP Manager;

– test SNMP functionality: These tests involve utilizing specific commands to establish a connection with the device, retrieve information from it, and execute administrative tasks. By carrying out these actions, the manager can assess whether the SNMP Agent is functioning correctly and the configuration is working as intended (Fig. 5, 6).

```
1 Switch(config)# snmp-server community NMS rw
2 Switch(config)# snmp-server location NMS_UI
3 Switch(config)# snmp-server contact ahmed@alhilali.com
4 Switch(config)# snmp-server enable traps
5 Switch(config)# ip access-list standard SNMP_ACL
6 Switch(config-std-nacl)# permit 172.16.0.2
```

Fig. 4. Simple Network Management Protocol Configuration

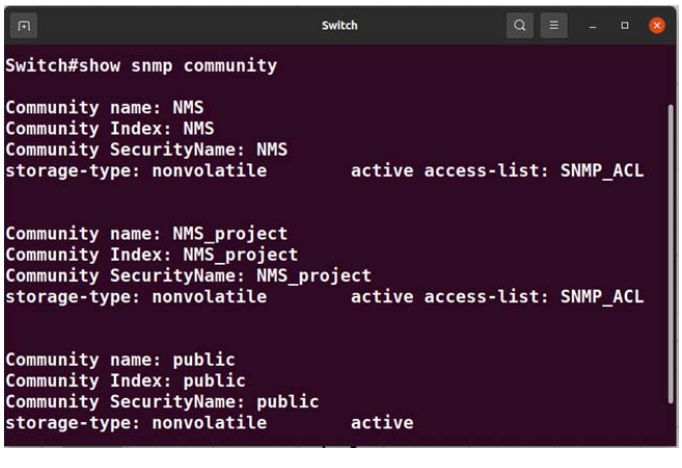

Fig. 5. The Community information

SNMP Manager offers network administrators an easy-to-use interface for interacting with SNMP Agents and managing network devices. In our work, the system is placed in the server to control the incoming traffic from all hosts and

disconnect the port of each host that exceeds a specific percentage of the total port capacity. In the Fig. 7, the snmpwalk command is used to get all the ifInOctete values for all switch ports with OID (.1.3.6.1.2.1.2.2.1.10) and the community string (NMS) and the Switch IP address (172.16.0.30).

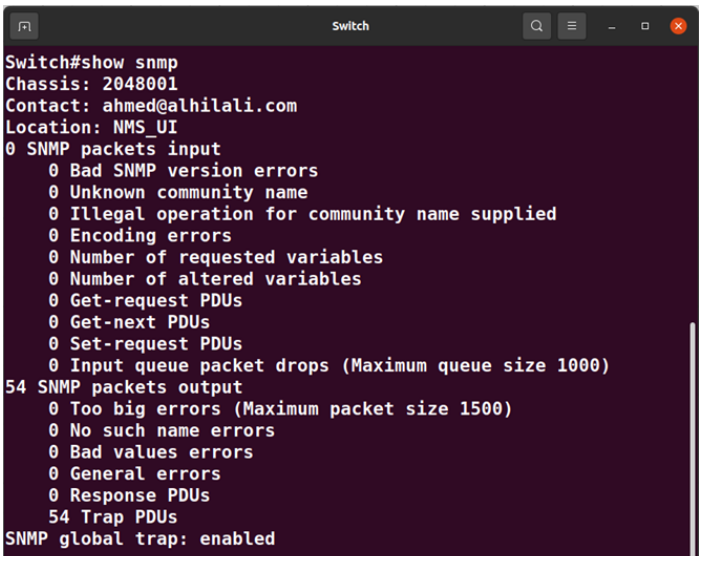

Fig. 6. Simple Network Management Protocol Configuration

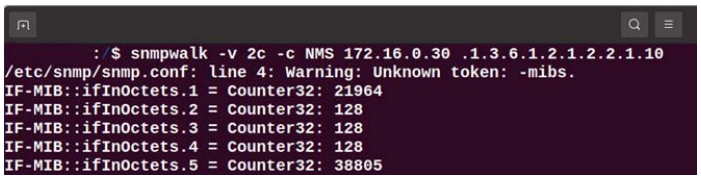

Fig. 7. Snmpwalk command output

The system functions are written in Python, and the PureSNMP library is used to get port information. This library enables the system administrator to query and manage network devices and easily retrieve vital information such as system status, network performance metrics, and device configuration data. The system will use SNMP GET to retrieve a port's information, and the SNMP SET to disable/enable the port.

5. 2. Development of an algorithm for implementing the system

To calculate the port incoming traffic, two values (ifSpeed and ifInOctets) are needed for each port. According to [25], ifSpeed represents an approximation of the current bandwidth of the interface, expressed in bits per second. And its OID is ({iso(1) identified-organization(3) dod(6) internet(1) mgmt(2) mib-2(1) interface(2) ifTable(2) ifEntry(1) ifSpeed(5)}). Also, ifInOctets refers to the cumulative count of octets received on the interface, encompassing framing characters, and its OID ({iso(1) identified-organization(3) dod(6) internet(1) mgmt(2) mib-2(1) interface(2) ifTable(2) ifEntry(1) ifInOctets(10)}).

Let's use the *IfInOctets*2 and *IfInOctets*1 variables in (1) to calculate the incoming port traffic (T) between two specific time points. IfInOctets2 represents the total incoming traffic at the later point in time, while IfInOctets1 represents the incoming traffic at the earlier point. To convert the traffic measurement from bytes to bits, let's multiply the difference between *IfInOctets*2 and *IfInOctets*1 by 8. Moreover, to determine the traffic rate, let's divide this value by the time interval between the two measurements (Ti). This calculation lets's obtain the incoming traffic rate in bits per second:

$$T=\left(IfInOctets2-IfInOctets1\right)*8/Ti. \quad (1)$$

Finally, as illustrated in Fig. 8, the system will verify whether the value of T surpasses the predefined percentage set by the network administrator. The system automatically uses set commands to disable and enable port that exceeds the traffic-specific rate, as shown in Fig. 9.

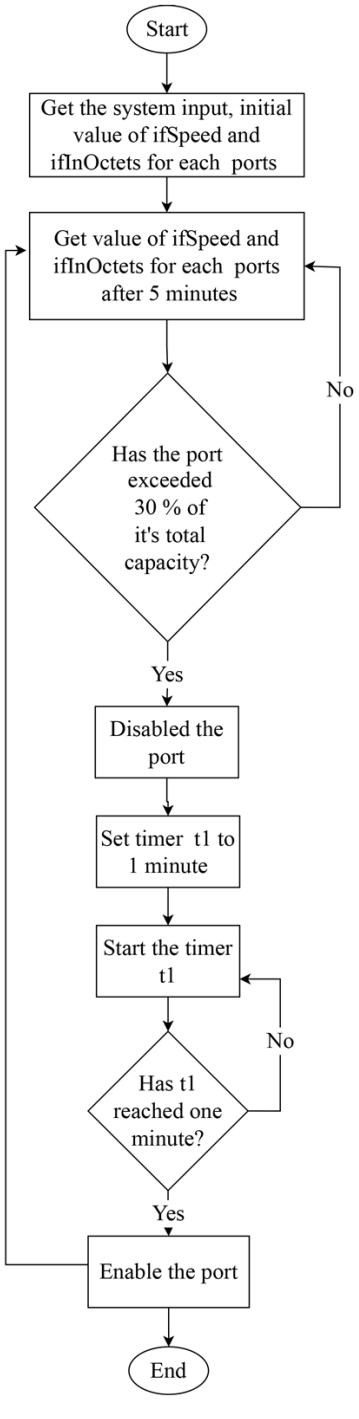

Fig. 8. Algorithm for implementing the system

```
set('172.16.0.30','NMS','1.3.6.1.2.1.2.2.1.7.2',Integer(2))
set('172.16.0.30','NMS','1.3.6.1.2.1.2.2.1.7.2',Integer(1))
```

Fig. 9. IfAdminStatus set command

The SNMP object identifier (OID) ifAdminStatus will be used by the system to monitor and manage the administrative status of a network interface on a device, which indicates the desired condition or state of the interface [26]. It includes three statuses:

– the testing state is indicated with the number 3.
– the downstate is indicated with the number 2.
– the upstate is indicated with 1.

5. 3. The system evaluation

The proposed system works in real time to detect the server's high traffic overload. The server involves the following specifications (OS: Linux Ubuntu 20.04, CPU: Core i7 3600, RAM 16 GB). Let's consider two elements (CPU and bandwidth usage) to evaluate the system's efficiency. The starting condition of the server is captured to compare with the overloaded ones, as depicted in Fig. 10. In this state, the server runs with minimal services and applications. The CPU usage remains within the 8 % range, indicating low utilization. Additionally, the bandwidth consumed by users a maximum of 260 bytes/s when sending and receiving data.

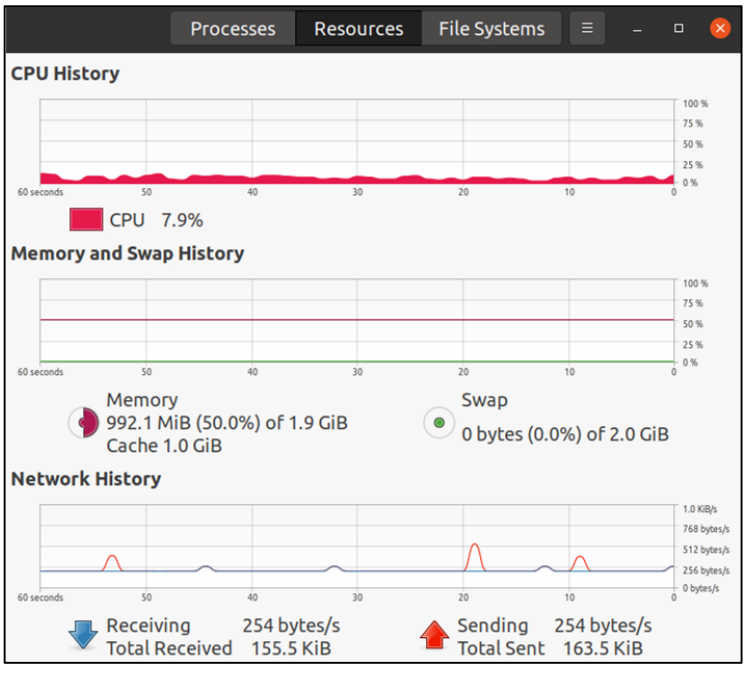

Fig. 10. Server resources usage in normal state

The system is written in Python and executed on the server through a Linux terminal. It will alert the network manager if the incoming traffic exceeds 30 % of the total port capacity and which port is disabled. When the traffic generated by the traffic generator through one of the hosts is unexpected, the port interface will be downstate for one minute. A ping command is used to verify the port status, as illustrated In Fig. 11.

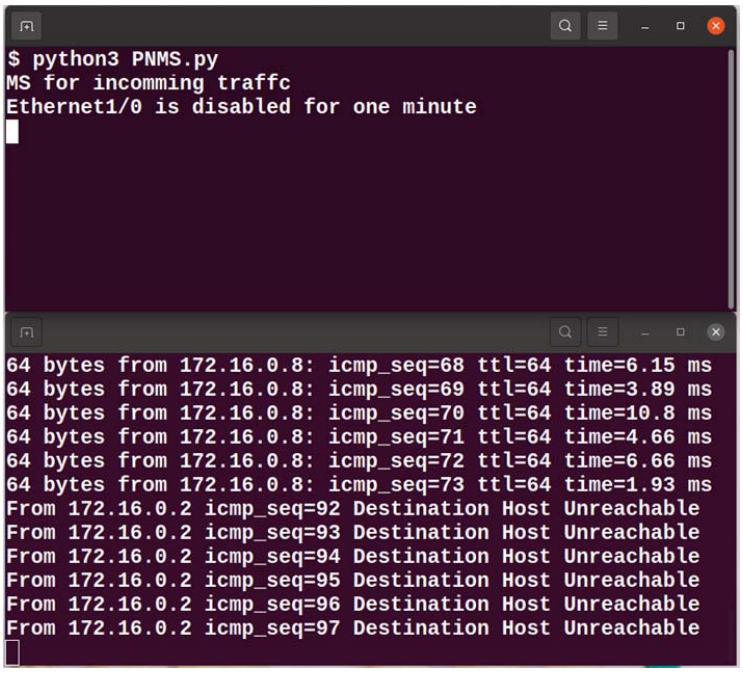

Fig. 11. System manager interface

In contrast, when the system is inactive, as depicted in Fig. 12, it exhibits a condition of traffic overload. During this period, the CPU usage significantly increased, reaching 28.1 %. Simultaneously, the bandwidth utilization experienced a sharp rise, exceeding 1.7 MiB/s (the receiving data). This comparison underscores the noteworthy shift in system performance and resource utilization between the active and inactive states, as the depicted figures show.

As illustrated in Fig. 13, when the system identifies a surge in network traffic exceeding predefined thresholds (30 % of the port's total capacity), it responds swiftly by deactivating the designated port for one minute. This proactive measure serves as a protective mechanism to maintain the overall health and stability of the system. Consequently, this action leads to restoring all resource values to their initial, optimal state. By promptly and effectively managing high-traffic scenarios in this manner, the system ensures the preservation of its performance and resource utilization, enhancing its reliability and efficiency.

Our proposed system plays a crucial role in maintaining network stability and server performance by vigilantly monitoring incoming network traffic. It operates with precision, constantly evaluating the data flow against meticulously configured thresholds, which are thoughtfully defined by the system administrator. This meticulous scrutiny ensures that the network remains well within the bounds of its capacity, thereby safeguarding it from the perilous brink of overload.

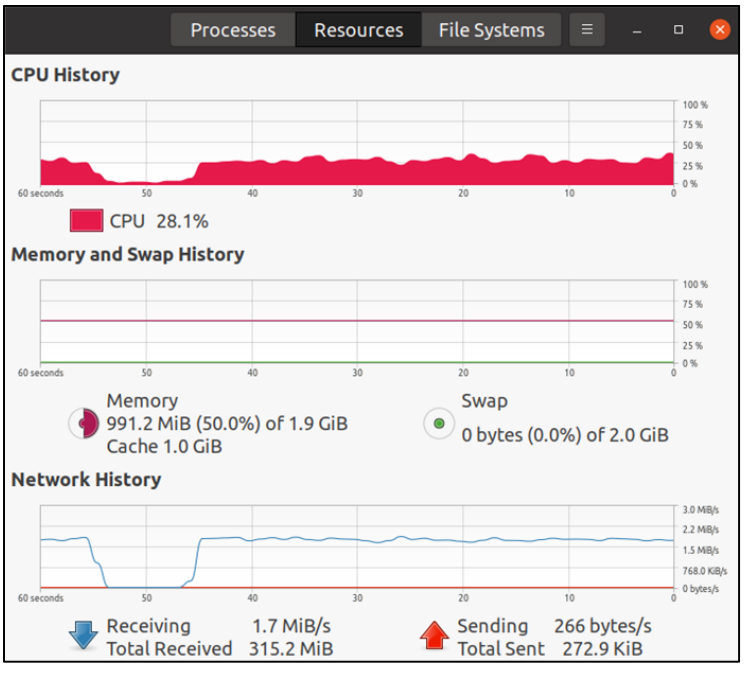

Fig. 12. Server resources usage in overload state

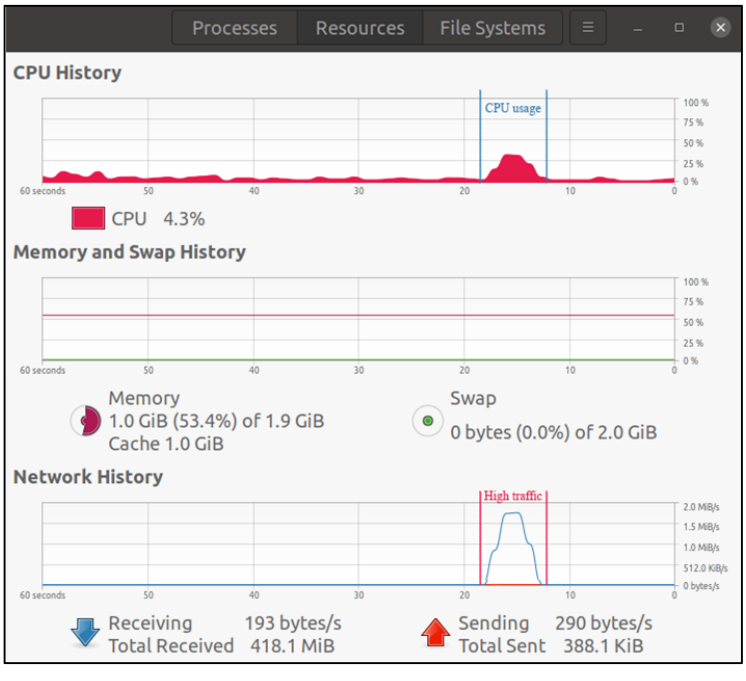

Fig. 13. Resources usage while the system is enabled

6. Discussion of results on the design and implement a real-time network traffic management system using SNMP protocol

In this article, it is developed a system that leverages the SNMP protocol to monitor and control incoming server traffic. This proposed method empowers network administrators to efficiently manage their networks, addressing the following key research objectives:

– describing the system design, illustrating the managed network topology, and configuring each connected device. Detailed information is presented in Fig. 3 and Table 1;

– introducing an algorithm for implementing the management method, enabling the system to gather essential information about all network devices and make decisions regarding port disabling for high bandwidth usage, as depicted in Fig. 8;

– conducting a system evaluation process, which involved subjecting the system to high traffic from each device to assess its functionality. Fig. 10–13 highlights the system in action.

The results demonstrate the system's capability to reduce CPU usage and bandwidth consumption from 28.1 % (CPU) and over 1.7 MiB/s (bandwidth) to the initial server stage, achieving 8 % CPU usage and 260 bytes/s bandwidth. In our future work, let's aim to enhance the system by incorporating additional functionalities and implementing a graphical user interface (GUI) to display the monitoring results. Furthermore, the network traffic data holds the potential for examining network behavior and identifying questionable actions, such as a Distributed Denial of Service (DDoS) assault. The limitations of this approach are contingent on the presence of physical devices to construct intricate networks and evaluate the system under various conditions. Conversely, the proposed solution may present a drawback when implemented in large-scale networks. To mitigate this issue, network administrators need to reconfigure the network by increasing the count of SNMP agents and managers to evenly distribute the workload.

7. Conclusions

1. An innovative system has been created with the ability to effectively manage incoming server traffic while preserving server workload and service performance. The developed system uses the SNMP protocol to monitor and control the network devices without needing a third-party tool. Additionally, it incorporates a rapid notification mechanism to alert administrators about high traffic levels and can turn off the responsible port. Furthermore, this system simplifies the management process, requires minimal computational resources, and assures the requisite level of performance by reducing CPU and bandwidth consumption.

2. The algorithm for implementing the system was defined, enabling the following functionalities:

– gathering information about all switch ports;

– checking if the incoming traffic (utilization) on switch ports exceeds a specific percentage of their total capacity;

– temporarily disabling ports that exceed the specified capacity percentage for a specific duration using the SNMP command "set ifAdminStatus" with Integer(2). Notably, our proposed algorithm stands out by utilizing SNMP commands for monitoring and controlling network devices.

3. Implementation results demonstrated the system's ability to swiftly deactivate designated ports upon detecting high traffic, restoring all resource values to their initial states. In contrast, during periods of traffic overload where CPU usage reached 28.1 %, and bandwidth utilization exceeded 1.7 MiB/s, our proposed system reduced these values to a CPU usage of 8 % and bandwidth consumption capped at 260 bytes/s.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Manuscript has no associated data.

Acknowledgment

We express our gratitude for the assistance provided by the Information Technology Research and Development Centre at the University of Kufa during the course of this research.

References

1. Saarikko, T., Westergren, U. H., Blomquist, T. (2017). The Internet of Things: Are you ready for what's coming? Business Horizons, 60 (5), 667–676. doi: https://doi.org/10.1016/j.bushor.2017.05.010
2. Mjhool, A. Y., Alsabah, R., Aljshamee, M., Alajwadi, M., Al-Sabbagh, A. (2021). E-learning Trends and Internet challenges in Iraq. International Journal of Computing and Digital Systems, 10 (1), 1407–1414. doi: https://doi.org/10.12785/ijcds/1001124
3. Aitymova, A., Shaporeva, A., Kopnova, O., Kushumbayev, A., Aitymov, Z. (2022). Development and modeling of combined components of the information environment. Eastern-European Journal of Enterprise Technologies, 2 (2 (116)), 51–60. doi: https://doi.org/10.15587/1729-4061.2022.255084
4. Wairisal, M., Surantha, N. (2018). Design and Evaluation of Efficient Bandwidth Management for a Corporate Network. 2018 International Conference on Information Management and Technology (ICIMTech). doi: https://doi.org/10.1109/icimtech.2018.8528162
5. Namee, K., Vantaneeyakul, C., Polpinij, J., Albadrani, G. M., Namee, S. (2020). Integration of Cloud Computing with Internet of Things for Network Management and Performance Monitoring. 2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE). doi: https://doi.org/10.1109/ictke50349.2020.9289876
6. Chris, H. (2019). What's Not So Simple about SNMP? Information Security Management, 93–106. doi: https://doi.org/10.1201/9781351073547-8
7. S, K. E., A, L. A. S. I., Widiartha, I. B. K. (2016). Perancangan Network Monitoring Tools Menggunakan Autonomous Agent Java. Lontar Komputer : Jurnal Ilmiah Teknologi Informasi, 7 (2), 115–121. doi: https://doi.org/10.24843/lkjiti.2016.v07.i02.p05
8. Miftah, Z. (2019). Penerapan sistem monitoring jaringan dengan protokol snmp pada router mikrotik dan aplikasi dude studi kasus stikom cki. Faktor Exacta, 12 (1), 58. doi: https://doi.org/10.30998/faktorexacta.v12i1.3481
9. Buchyk, S., Yudin, O., Ziubina, R., Bondarenko, I., Suprun, O. (2021). Devising a method of protection against zero-day attacks based on an analytical model of changing the state of the network sandbox. Eastern-European Journal of Enterprise Technologies, 1 (9 (109)), 50–57. doi: https://doi.org/10.15587/1729-4061.2021.225646
10. Majidha Fathima, K. M. (2021). A Survey of the Exemplary Practices in Network Operations and Management. Algorithms for Intelligent Systems, 181–194. doi: https://doi.org/10.1007/978-981-15-8530-2_14
11. Espinel Villalobos, R. I., Ardila Triana, E., Zarate Ceballos, H., Ortiz Triviño, J. E. (2021). Design and Implementation of Network Monitoring System for Campus Infrastructure Using Software Agents. Ingeniería e Investigación, 42 (1), e87564. doi: https://doi.org/10.15446/ing.investig.v42n1.87564
12. Falih Kadhim, M., Al-Janabi, A., Hazim Alhilali, A., Salih Ali, N. (2022). Security approach for instant messaging applications: viber as a case study. Indonesian Journal of Electrical Engineering and Computer Science, 26 (2), 1109. doi: https://doi.org/10.11591/ijeecs.v26.i2.pp1109-1115
13. Rasyiidin, M. Y. B., Murad, F. A., Murad, F. A. (2021). Monitoring Server Berbasis SNMP Menggunakan Cacti pada Server Lokal. Jurnal Ilmiah FIFO, 13 (1), 14. doi: https://doi.org/10.22441/fifo.2021.v13i1.002
14. Matoušek, P., Ryšavý, O., Polčák, L. (2021). Unified SNMP Interface for IoT Monitoring. In 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), 938–943. Available at: http://dl.ifip.org/db/conf/im/im2021-ws3-manage-iot/213211.pdf
15. Li, H. (2019). Design and Implementation of Spacecraft Network Monitoring System based on SNMP. 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). doi: https://doi.org/10.1109/imcec46724.2019.8983796
16. Da Costa, E., Mesquita, S. (2022). Computer Network Management and Monitoring System With SNMP and QoS Approach. Timor-Leste Journal of Engineering and Science, 3 (1), 19–27. Available at: https://tljes.org/index.php/tljes/article/download/30/13

17. Barannik, V., Sidchenko, S., Barannik, N., Barannik, V. (2021). Development of the method for encoding service data in cryptocompression image representation systems. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 103–115. doi: https://doi.org/10.15587/1729-4061.2021.235521
18. Madi, N., Al-kasassbeh, M. (2019). Collecting MIB data from network managed by SNMP using multi mobile agents. arXiv. doi: https://doi.org/10.48550/arXiv.1909.02547
19. Safrianti, E., Sari, L. O., Sari, N. A. (2021). Real-Time Network Device Monitoring System with Simple Network Management Protocol (SNMP) Model. 2021 3rd International Conference on Research and Academic Community Services (ICRACOS). doi: https://doi.org/10.1109/icracos53680.2021.9701973
20. Sheeraz, M., Paracha, M. A., Haque, M. U., Durad, M. H., Mohsin, S. M., Band, S. S., Mosavi, A. (2023). Effective Security Monitoring Using Efficient SIEM Architecture. Human-centric Computing and Information Sciences, 13. doi: https://doi.org/10.22967/HCIS.2023.13.023
21. Valencic, D., Mateljan, V. (2019). Implementation of NETCONF Protocol. 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). doi: https://doi.org/10.23919/mipro.2019.8756925
22. Maulana, F. (2016). Implementasi Simple Network Management Protocol (Snmp) Pada Aplikasi Monitoring Jaringan Berbasis Website(Studi Kasus Universitas Muhammadiyah Bengkulu). Jurnal Informatika, 16 (2), 126–135. Available at: https://jurnal.darmajaya.ac.id/index.php/JurnalInformatika/article/view/947/pdf
23. The SNMP architecture. IBM. Available at: https://www.ibm.com/docs/en/informix-servers/14.10?topic=concepts-snmp-architecture
24. Nidhishree, G., Manimala, S. (2017). Design and Implementation of SNMP based Network Device Monitoring System. International Journal of Trend in Research and Development (IJTRD), 4 (3), 383–385. Available at: http://www.ijtrd.com/ViewFullText.aspx?Id=9632
25. Hammood, N. H., Al-Musawi, B., Alhilali, A. H. (2022). A Survey of BGP Anomaly Detection Using Machine Learning Techniques. Communications in Computer and Information Science, 109–120. doi: https://doi.org/10.1007/978-981-19-1166-8_9
26. Global OID reference database. Available at: https://oidref.com/
27. SNMP MIB. Oracle. Available at: https://docs.oracle.com/cd/E13203_01/tuxedo/tux91/snmpmref/1tmib.htm
28. Lee, H.-J. (2023). Dynamic Context Awareness of Universal Middleware based for IoT SNMP Service Platform. Tehnički Glasnik, 17 (2), 185–191. doi: https://doi.org/10.31803/tg-20221221115431
29. Simulating Network Architectures with GNS3 (2020). Systems and Network Infrastructure Integration, 9–25. doi: https://doi.org/10.1002/9781119779964.ch2