Implementation of Network Monitoring using Mobile Push Notification Systems

E Handoyo[1], FR Adhipratama[2], M Somantri[3], Susatyo Handoko[4], Karnoto[5], Tejo Sukmadi[6], Sukiswo[7], Sudjadi[8]

Department of Electrical Engineering,
Undip, Central Java, Indonesia
arfan.undip@gmail.com

Abstract—*The industrial era 4.0 brings significant changes to people's digital habits. Industry 4.0 changes the paradigm of the Internet of People to become the Internet of Things. The change in the paradigm of the Internet of People to become the Internet of Things also affects communication and data network systems. Network management takes care of maintenance and development. In monitoring network performance using Zabbix servers. The data transmission protocols used are SNMP and ICMP protocols. Data sent by intermediary devices using SNMP and ICMP protocols will be stored on Zabbix servers. Zabbix servers are used as a platform for storing data in the form of hosts, traffic, logs, interconnection between interfaces, and so on. If the Zabbix server detects a broken port connection and high traffic, then the alert notification will be sent to telegram. With this system, network engineers can monitor network topology conditions in real-time and remotely.*

Keywords—Zabbix Server; Telegram Alert; SNMP Protocol; ICMP Protocol

1. INTRODUCTION

The industrial era 4.0 brings major changes in network and data communication systems. The industrial era 4.0 allows communication between non-living devices. The process of data transfer activities between these devices needs to be arranged in a network management system. Network management is the ability to monitor, control and plan resources and components of computer and network systems [1]. In the process of transmitting digital data via Ethernet or wireless, it is not uncommon to find connectivity problems between the two nodes. Problems encountered can include broken connections between two nodes or traffic that is too high. To be able to prevent this, technicians generally periodically ping the associated port to check the port's connectivity. Cacti server-based monitoring system to improve network management effectiveness. The parameters monitored include port connectivity (up or down) as well as upload and download traffic rates. However, obstacles were found in notification management. As a solution, the authors apply a network monitoring system using the Zabbix server as a network monitoring log data storage medium. In supporting the effectiveness of network management, alerts are used in the form of Telegram notifications. The Telegram bot will send a message when the system detects that the port connectivity is down or very high traffic.

2. METHODOLOGY

2.1 Network Monitoring System

Monitoring Network is a function of management that is useful for analyzing whether the network is still suitable for use or needs additional capacity. Monitoring results can also help if the admin wants to redesign the existing network. Many things on the network can be monitored, one of which is the load of network traffic passing through a router or computer interface. Monitoring can be done with SNMP standards, in addition to network traffic load, network conditions must also be monitored, for example the up or down status of a network equipment. This can be done with the ping utility.[2]

A monitoring system performs the process of collecting data about itself and analyzing these data in order to maximize all its resources. Generally, the data collected is real-time data, both data obtained from hard real-time systems and soft real-time systems. A real-time system is a system where the time required by a computer to provide a stimulus to the external environment is vital. Time in this sense means that a real-time system runs a job that has a deadline [3].

Broadly speaking, the stages in a monitoring system are divided into three major processes, namely the process in collecting monitoring data; Process in monitoring data analysis and Process in displays data the result monitoring.

2.2 Simple Network Management Protocol

In simple terms, SNMP is a protocol designed to provide the ability for users to manage their computer networks remotely or remotely. The SNMP protocol provides a mechanism for management entities, or stations, to extract information from the Management Information Base (MIB) of managed devices [4]. This management is carried out by conducting polls and setting the variables of the network elements it manages. MIB or Management Information Base, can be said to be a database structure of managed element variables. This structure is hierarchical and has rules in such a way that information on the value of each variable can be known or set easily. Agent is software that runs on each node or network element to be monitored. His job is to collect all the information that has been determined in the MIB. Manager is software that runs on a host on a network. This manager is in charge of collecting information from agents. Not all information held by the agent is requested by the manager. The information requested by the network administrator, who runs the host that functions as manager, will be collected from the agent.

PDU (Protocol Data Unit) is a data unit that consists of a header and some embedded data. Viewed from the above perspective, this PDU can be seen as an object that contains variables. This variable has a name and a value.

The SNMP protocol uses relatively simple operations and a limited number of PDUs to perform its functions. The five PDUs defined in the standard are as follows:

a. Get Request: This PDU is used to access the agent and get values from the list of variables requested. This PDU contains an identifier that differentiates it from multi requests or variable values (network element status).

b. Get-Next Request: Like Get Request, but allows retrieval of information on the next logical identifier in the MIB tree sequentially.

c. Get Response: This PDU is to respond to the Get Request, Get-Next Request, and Set Request data units, so it is issued by the agent.

d. Set Request: Used to describe an action to be performed on a network element. Usually to change the value of a variable list.

e. Trap: This PDU allows the network management / agent module to report events on network elements to the manager.

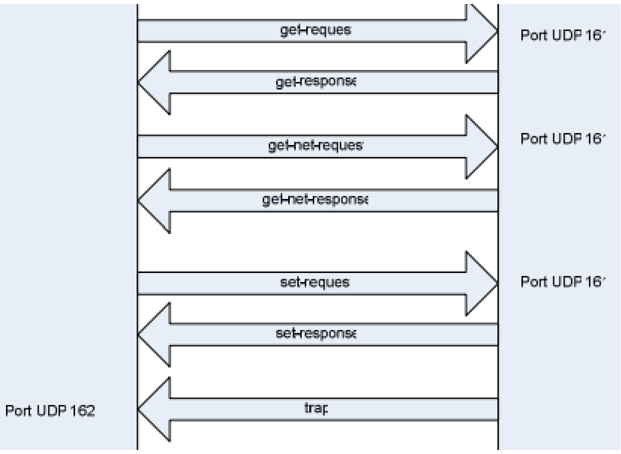


Figure 1: SNMP operator

2.3 Internet Control Message Protocol and Network Operating System

ICMP (Internet Control Message Protocol) is a protocol that is tasked with sending error messages and other conditions that require special attention. ICMP messages / packets are sent if there is a problem at the IP layer (layer 3) and the upper layer (TCP / UDP) (layer 4). In normal condition, the IP protocol works fine. However, there are several conditions where the IP connection is interrupted, for example because the router crashes, disconnection of the cable, or death of the destination host. At this time ICMP helps stabilize network conditions, by providing certain messages in response to certain conditions that occur on the network.[5]

Network Operating System (NOS) is software that runs on a server and allows the server to manage data, users, groups, security, applications, and other network functions. The Network Operating System (NOS) is based on a client / server architecture where the server allows multiple clients to share resources. Network Operating System (NOS) distributes its functions over a data communication network. The NOS operating system depends on each individual computer's native OS. Then add a function that allows access to resources shared by the number of users concurrently [6].

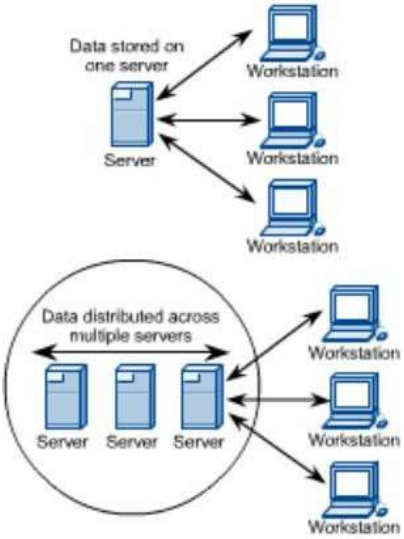


Figure 2: Simple Architecture of Network Operating System

Figure 2 shows that the NOS server is a multitasking system. Multitasking system means that internally, the OS must be able to run several tasks or processes at the same time. The operating system performs scheduling software that is built into the execution environment. Scheduling software allocates internal processor time, memory, and other elements of the system to different tasks in ways that allow them to share system resources. Each user on a multiuser system is supported by a separate task or process internally on the server.

2.4 Zabbix Server

Zabbix is software that monitors various network parameters, server health, and integrity. Zabbix uses a flexible notification mechanism that allows users to configure email based alerts for most events. This allows for quick reactions to server problems. Zabbix offers excellent reporting and data visualization features based on stored data. This makes Zabbix ideal for capacity planning. Zabbix supports polling and trapping. All Zabbix reports and statistics, as well as configuration parameters are accessed via the web-based front end. Zabbix can play an important role in monitoring the IT infrastructure. This is also true for small organizations with few servers and for large companies with many servers [7].

Figure 3: *Server Dashboard* Zabbix

To facilitate the network monitoring process, the Zabbix server has a dashboard with an attractive graphical user interface (GUI). The dashboard page displays problem notifications from each host. The dashboard page also displays system information. Each information is obtained from log data collected by Zabbix servers. The Zabbix server is used as a storage medium and intranet network traffic log data is sent to Zabbix servers via SNMP protocol on Cisco devices. [8]

3. Result

3.1 Server Configuration

The implementation of the Cisco network monitoring system in this case uses the Zabbix server version 5.2.4. The Zabbix server acts as a virtual machine that is installed on the Ubuntu server version 18.04. In testing, used 2 servers with different minimum systems.

The first server used has 12GB of RAM memory which is divided into two memory slots. The memory bank 0 slot uses Kingstone RAM with a clock of 1600MHz. Memory bank slot 1 uses Micron RAM with a clock of 1866MHz. Supported by a 500 GB ATA Disk vendor Western Digital 7200 rpm. The first server runs on a quad core AMD A8-8650B processor with a maximum clock rate of 3200 MHz. In the Zabbix server configuration, the server is set with a cache memory capacity of 2 GB to store Zabbix server discovery logs against Cisco devices.

The second server used has 32GB of RAM memory which is divided into two memory slots. Memory bank slot 1 uses Kingston DDR4 RAM with a clock of 2666MHz. Memory bank slot 3 uses Kingston RAM with a clock of 2666MHz. Supported by storage space with a solid state disk of 1000 GB. The solid state disk used is the Samsung SSD 870 ATA Disk type with a write speed of 530 MBps and a read speed of 560 MBps. The second server runs on an 8-core Intel Core i7-10700 processor with a maximum clock rate of 4800MHz. In the Zabbix server configuration, the server is set with a cache memory capacity of 8 GB for storage. Zabbix server discovery logs of Cisco devices.

3.2 Host and Notification Configuration

Host is a device that will be monitored by the Zabbix server. In this case the 193 hosts monitored by the Zabbix servers were used. Judging from the type of interface setting, the host is divided into 2 interfaces, SNMP and Agent. For switches and routers using the SNMP protocol. Meanwhile, for server equipment using an agent.

The purpose of implementing notification alerts through telegram chat bots is to provide emergency notifications to the admin if there is downtime or high traffic to the agent. This system works in conjunction with the polling system. Poller is a Zabbix server process which is responsible for fetching data from Zabbix and SNMP agents and processing (simple) remote checking [9]. If the agent's discovery result states that the agent is experiencing downtime and / or low speed due to very high traffic, then an alert notification will be sent to telegram via a chat bot.

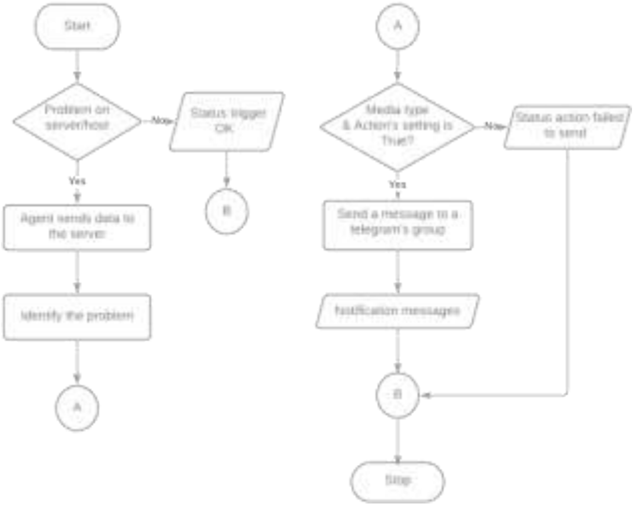

Start
Problem on server/host
No
Status trigger OK
B
Yes
Agent sends data to the server
Identify the problem
A
A
Media type & Action's setting is True?
No
Status action failed to send
Yes
Send a message to a telegram's group
Notification messages
B
Stop

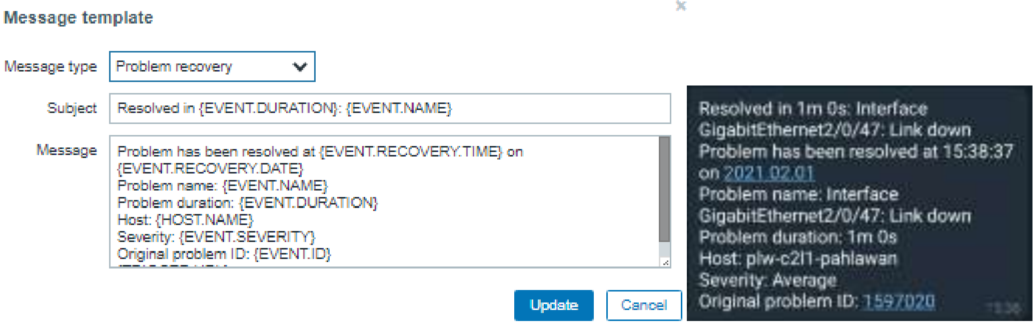
Figure 4: Alert Notification Delivery Flowchart Telegram

Figure 4 shows the workflow of sending alert notifications via Telegram. Poller is in charge of retrieving data from Zabbix and SNMP agents. When the Zabbix server receives problem data from the poller, the server identifies the related problem. The problem will be classified by the server into several criteria as shown in table 1

Table 1: Host Problem Criteria

Classification	Description	Color Notofication
Not Classified	Problem not classified	Gray
Warning	Notification character warning.	Yellow
Average	Problem character medium.	Dark red
High	Problem character dangerous.	Red
Disaster	Problem character disaster, as lost data.	Bright red

In sending notification alerts, Zabbix servers use message templates as a draft message sent via telegram. Templates are divided into 2 types of messages, namely Problem and Problem Recovery. Message Problem contains problems with the associated port interface, host name, severity or problem classification, operational data, and problem ID.[10]



Message template

Message type: Problem recovery

Subject: Resolved in {EVENT.DURATION}: {EVENT.NAME}

Message:

```
Problem has been resolved at {EVENT.RECOVERY.TIME} on
{EVENT.RECOVERY.DATE}
Problem name: {EVENT.NAME}
Problem duration: {EVENT.DURATION}
Host: {HOST.NAME}
Severity: {EVENT.SEVERITY}
Original problem ID: {EVENT.ID}
```

```
Resolved in 1m 0s: Interface
GigabitEthernet2/0/47: Link down
Problem has been resolved at 15:38:37
on 2021.02.01
Problem name: Interface
GigabitEthernet2/0/47: Link down
Problem duration: 1m 0s
Host: plw-c2l1-pahlawan
Severity: Average
Original problem ID: 1597020
```

Figure 5: Format of Problem Recovery Notification

4. Discussion

System testing focuses on server performance in Zabbix and message notifications. Performance testing is done by calculating the delay time between problems that appear on the Zabbix server log and messages received on the Telegram.

4.1 Notification Delivery

System testing focuses on server performance in Zabbix and message notifications. Performance testing is done by calculating the delay time between problems that appear on the Zabbix server log and messages received on the Telegram.

Table 2: Notification Delivery Test

Host	Test Date	Delay	Information
plw-c2l1- hero	February 1, 2021	0: 00 ': 44 "	Interface GigabitEthernet2/0/19 : Link down
plw-c2l3- hero	February 1, 2021	0: 13 ': 19 "	Interface GigabitEthernet2/0/26 : Link down
plw-c2l1- hero	February 2, 2021	2: 17 ': 57 "	Interface GigabitEthernet2/0/29 : Link down
pkl-c1l2-is	February 5, 2021	0: 03 ': 55 "	Interface GigabitEthernet1/0/18 : Link down

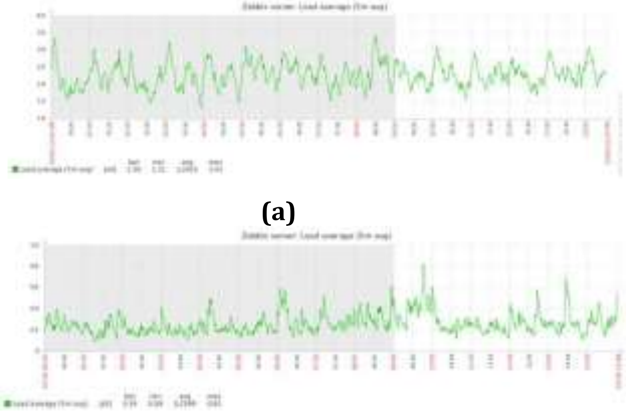
pkl-c1l3- sto-pkl	February 5 2021	0: 12 ': 54 "	*Interface* GigabitEthernet1/0/38 : Link down
tgg-c1l1- tegalplasa	February 7th 2021	3: 50 ': 54 "	*Interface* GigabitEthernet1 / 0/10 : Link down
plw-c1l2- hero	February 8th 2021	0: 00 ': 23 "	Gigabit Ethernet Interface 1/0/8: Link down

Table 2 shows the test results of sending Telegram notifications. The test was conducted on an Ubuntu server with a quad core AMD A8-8650B processor with a maximum clock rate of 3200 MHz supported by 500 GB of hard disk storage. The hard drive used is the Western Digital vendor's ATA Disk type 7200 rpm. It is calculated that the average delay value is 0:55:51.

Judging from the results of the notification delay test, it can be seen that the first server produces a delay of 55 minutes 51 seconds. Meanwhile, the notification delay test on the second server resulted in a delay of 11 seconds. This is because when testing the notification delay on the first Zabbix server, the Zabbix server service was shutdown. It is assumed that this is due to the following factors, in terms of CPU and the rate of write and read performance on the disk. The AMD A8-8650B quad core processor already has a core of 4 cores, broadly speaking it is capable of running Zabbix servers. The processor has a CPU thread of 4 threads. In terms of multitasking, this is not enough. In terms of storage, the server uses a 500GB HDD with a rpm rate of 7200rpm. In terms of performance, querying data and host discovery processes are deemed inadequate so that delays arise in the process.

4.2 Zabbix Load Server

The load average on a Linux server for a certain period of time. In other words, load average is the CPU request from the server which includes the number of threads running and waiting [11]. The average load is relative to the number of cores available on the server. This means that the maximum utilization range is 0-1 for a single core, 0-2 for a dual core, 0-4 for a quad core, 0-8 for an octa core (octa). core), and so on.



(a)

(b)

Figure 6: Graph of Zabbix Server Load Average

The first server was running on a quad core AMD A8-8650B processor with a maximum clock rate of 3200 MHz. This means that the first server is capable of running 0 to 4 loads on the system at one time. Based on Figure 4.16 (a), it can be seen that the average load value is 2.2659 processes, the maximum load value is 3.43 processes, and the minimum load value is 1.32 processes. The load average graph shows that the process being carried out is still able to be executed by the server because there has not been an overload. It can be seen that the average load value is still below 4. However, be aware that the maximum load value is 3.43 which means that at any time the server can run slowly.

The second server runs on an Ubuntu Server processor running on an 8 core Intel Core i7- 10700 processor with a maximum clock rate of 4800MHz. This means that the first server is capable of running 0 to 8 loads on the system at one time. Based on Figure 4.16 (b), it can be seen that the average load value is 0.2669 processes, the maximum load value is 0.82 processes, and the minimum load value is 0.09 processes. The load average graph shows that the process being carried out is still able to be executed by the server because there has not been an overload. It can be seen that the load average value is still below 8.

5. Conclusion

Theoretically, a Zabbix server that monitors less than 500 devices, requires a CPU with as many as two cores, but in practice, we need a CPU with more than two cores supported by a large CPU clock rate and a hard drive with large write and read performance. Based on the results of the notification delivery delay test on the old server (the first server), it resulted in a delay of 55 minutes 51 seconds. After replacing the server, it results in a notification delivery delay of 11 seconds.Based on a review of load average values, Ubuntu servers with 8-core Intel Core i7-10700 processors with a maximum clock rate of 4800MHz are better able to handle the computational load of Zabbix servers compared to Ubuntu servers with quad core AMD A8-8650B processors with a maximum clock rate of 3200 MHz. Network Operating System (NOS) works by distributing its functions through a data communication network so that devices connected can access data and services from the Zabbix server installed on it.

In maintaining the performance of Zabbix servers, it is necessary to have regular cleaning related to logs and other junk files. Several companies has monitoring servers spread across several physical server devices. In managing each server, you should build a dashboard with login system protection, where the dashboard contains a landing page and a navigation menu to the associated server (such as the Link tree model).

6. References

[1] A. März, M. Lachner, C. G. Heumann, J. H. Schumann, and F. von Wangenheim, “How You Remind Me! The Influence of Mobile Push Notifications on Success Rates in Last-Minute Bidding,” *J. Interact. Mark.*, vol. 54, pp. 11–24, May 2021, doi: 10.1016/j.intmar.2020.08.002.

[2] J. Dhillipan, N. Vijayalakshmi, and S. Suriya, “Network Monitoring System Using Ping Methodology and GUI,” in *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, V. E. Balas, R. Kumar, and R. Srivastava, Eds. Cham: Springer International Publishing, 2020, pp. 13–22.

[3] S. Forti, M. Gaglianese, and A. Brogi, “Lightweight self-organising distributed monitoring of Fog infrastructures,” *Futur. Gener. Comput. Syst.*, vol. 114, pp. 605–618, Jan. 2021, doi: 10.1016/j.future.2020.08.011.

[4] C. T. Yang, S. T. Chen, J. C. Liu, Y. Y. Yang, K. Mitra, and R. Ranjan, “Implementation of a real-time network traffic monitoring service with network functions virtualization,” *Futur. Gener. Comput. Syst.*, vol. 93, pp. 687–701, Apr. 2019, doi: 10.1016/j.future.2018.08.050.

[5] M. Sakai, K. Takahashi, and S. Kondoh, “A Self-Adaptive Measurement Rate Control Method for an Agent-based Service Monitoring System,” in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2020, pp. 155–160, doi: 10.23919/APNOMS50412.2020.9237011.

[6] M. Meyer and M. Helfert, “Enterprise Architecture,” *Comput. Handbook, Third Ed.*, vol. 05, no. 1, pp. 25-1-25–16, 2014, doi: 10.1201/b16768-30.

[7] A. Mardiyono, W. Sholihah, and F. Hakim, “Mobile-based Network Monitoring System Using Zabbix and Telegram,” in *2020 3rd International Conference on Computer and Informatics Engineering (IC2IE)*, 2020, pp. 473–477, doi: 10.1109/IC2IE50715.2020.9274582.

[8] E. Jo and H. Yoo, “Implementation of Cloud Monitoring System Based on Open Source Monitoring Solution,” in *Software Engineering in IoT, Big Data, Cloud and Mobile Computing*, Springer, 2021, pp. 181–190.

[9] S. K. Shivakumar, “Web Performance Monitoring and Infrastructure Planning,” in *Modern Web Performance Optimization*, Springer, 2020, pp. 175–212.

[10] W. Chi and W. Zhou, “A Realtime Monitoring Method for Cluster System Running State Based on Network,” in *Journal of Physics: Conference Series*, 2019, vol. 1302, no. 2, p. 22070.

[11] D. Chahal, L. Kharb, and D. Choudhary, “Performance Analytics of Network Monitoring Tools,” *Int. Journalof Innov. Technol. Explor. Eng. June*, 2019.