Integrated Intelligent Agent for SNMP-Based Network Management System

Dung Ong Mau(✉)

Industrial University of Ho Chi Minh City, Faculty of Electronics Technology, Ho Chi Minh City, Vietnam
ongmaudung@iuh.edu.vn

Abstract. This paper proposes intelligent Simple Network Management Protocol (i-SNMP) agents, which add intelligent functions to the network routers, help routers know the most updated status of the whole network, and efficiently improve the performance as well as the stability of network devices. i-SNMP agents are able to monitor the network nodes, collect real-time network data, and communicate with the SNMP manager in the community. The SNMP manager can also diagnose and analyze the collected operational data and identify network anomalies. Then, proper actions can be taken to correct the network faults. OPNET Modeler is used in our research as the simulated network environment. By using OPNET Modeler, we describe the design and implementation of i-SNMP agents for managing the network and detecting network faults successfully.

Keywords: Network Management System · Simple Network Management Protocol · OPNET Modeler

1 Introduction

As the network becomes more complex, maintaining fast and reliable network communication becomes an important task for network management. Identifying network failures for effective network management is very important to ensure real-time performance. Routers play an important role in network management since they make decisions about which of several possible network paths data should follow, whether routers can arrive at the right decision will have a great impact on the network's performance [1]. But most of the routers on the market haven't sophisticated diagnosis abilities that can lead them to arrive at intelligent decisions, even for those intelligent routers. Routers are limited to global information about the network environment, only getting information from surrounding routers. One of the reasons is that as the router travels further into a geographic area, more data must be processed and analyzed, causing the router's speed to slow [2,3].

N.-S. Vo and H.-A. Tran (Eds.): INISCOM 2023, LNICST 531, pp. 19–33, 2023.
https://doi.org/10.1007/978-3-031-47359-3_2

There have been several discussions and implementations of intelligent agents. They pointed out that intelligent agents hold the most promise in bringing maverick equipment from outside organizations into the network-monitoring fold. There are various intelligent agent implementations, and most of the existing approaches emphasize the management of the network. These approaches could not solve the problem those routers have a limited view of the entire network environment [4–6]. Our goal is to get a global picture that can help the routers make more intelligent decisions about how to direct network traffic and at the same time avoid slowing down the speed of the router.

For this reason, we decided to build an intelligent agent, which sits beside the router and collects information from geographically dispersed servers using Simple Network Management Protocol (SNMP) application. It also should be able to analysis the information collected and provide the analysis result of the current status for network component to the router so that the router can make more intelligent decisions. We choose to use the SNMP application because SNMP is a widely used network management protocol. Another reason is that the connection-less communication feature and simple frame structure of SNMP have greatly promoted its applications. The two main components of SNMP (agents and manager) have independent operating mechanisms. Even though agents may fail and stop working, manager can continue to work normally.

In order to design and implement our proposed i-SNMP, an Ethernet LAN environment is created in OPNET Modeler [7,8]. Various kinds of traffic are implemented, and network fault scenarios are simulated. The i-SNMP agent is implemented on each router based on the SNMPv2 architecture. The SNMP manager is created to poll each agent and receive network data through SNMP packets. Data are processed, and analysis is done on the SNMP manager to detect network faults. The OPNET simulation results show that the i-SNMP agent is able to monitor the status of routers, identify network abnormalities such as heavy congestion and transient errors that are difficult to detect by network devices alone, and improve the performance of routers. The i-SNMP agent can be implemented on other network nodes, such as switches or servers. Because of the distributed agent architecture, the i-SNMP agent is able to monitor and manage networks at both small and large scales.

The remaining paper divides into the following primary sections: Sect. 2 highlights our proposed i-SNMP architecture. Section 3 describes the network system in OPNET Modeler in detail. Section 4 describes scenario simulations and results. Finally, the paper concludes in Sect. 5.

2 i-SNMP Architecture

The two key components in the project are the i-SNMP agent and SNMP agent manager. i-SNMP agents are responsible for monitoring the network nodes, collecting real-time network data, and communicating with the intelligent agent (SNMP manager) in the same community. SNMP manager receives information from the agents, analyzes the collected operational data, diagnoses the network fault, and reports the status of the network components [9].

2.1 Client and Server Architecture

The SNMP application is developed in a client/server architecture to implement SNMP version 2 and for network management. The SNMP application is based on the generic application model of OPNET Modeler, which models the detailed behaviors of the SNMP application [10,11].

The client and the server are the two major roles in the application architecture. In this paper, the agent manager is a workstation, which is running the SNMP application as a client, and the intelligent router is a network router, which is running the SNMP application as a server.

2.2 SNMP Message Exchange Architecture

The agent manager and the intelligent routers have different functions for network communication and data analysis. The agent manager is able to send *getRequest* packets to the router, receive *getResponse* packets, store the information in its database, analyze the information, and detect network faults. Similarly, the intelligent router is able to receive *getRequest* packets from the agent manager and send *getResponse* packets with its current information.

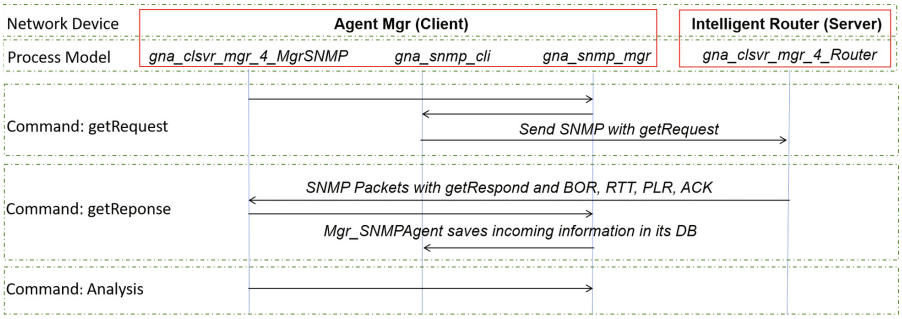


Fig. 1. Three scenarios of the SNMP application.

In order to illustrate the SNMP application for network management, we divided it into three scenarios. They are the *getRequest* scenario, the *getResponse* scenario, and the analysis scenario. Figure 1 illustrates these three scenarios for the SNMP application for network management. In the *getRequest* scenario, the agent manager sends *getRequest* packets to the intelligent routers to request their current status. In the *getResponse* scenario, the intelligent routers send the *getResponse* packets to the agent manager with their real-time status, such as Buffer Occupancy Rate (BOR), Round-Trip Time (RTT), Packet Loss Rate (PLR), and Acknowledgement (ACK). The agent manager receives *getResponse* packets and saves the information to its database. In the Analysis scenario, the agent manager analyzes the operational data of intelligent routers in time series

and detects the three kinds of network faults: congestion, equipment failure, and transient error.

To achieve these three scenarios of the SNMP application, four process models were created. The process models of *gna-clsvr-mgr4-MgrSNMP*, *gna-snmp-mgr*, and *gna-snmp-cli* are developed for the agent manager, and *gna-clsvr-mgr4-Router* is implemented for the intelligent router. The agent manager invokes three different process models to complete three scenarios, and the intelligent router invokes one process model to respond to the requests from the agent manager.

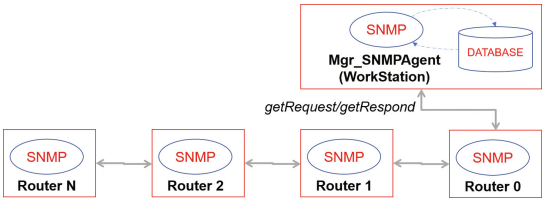


Fig. 2. SNMP message exchange mechanism and architecture.

Figure 2 illustrates the SNMP message exchange mechanism. The agent manager and the intelligent routers are running the SNMP application. They exchange SNMP messages by sending and receiving packets. The agent manager maintains a database for all the collected information from the intelligent agents.

This SNMP application is integrated into the network routers of OPNET Modeler to implement SNMP for network management. By adding these process models into the process architecture of routers, the routers can support SNMP. By running the SNMP applications on the agent manager and the intelligent routers, the agent manager can easily monitor the distributed intelligent routers, which are located in different subnets. And by analyzing the current information from intelligent routers, the agent manager can immediately detect four network faults. The network administrator and other applications can acquire the status of network routers in real time.

3 Network System in OPNET Modeler

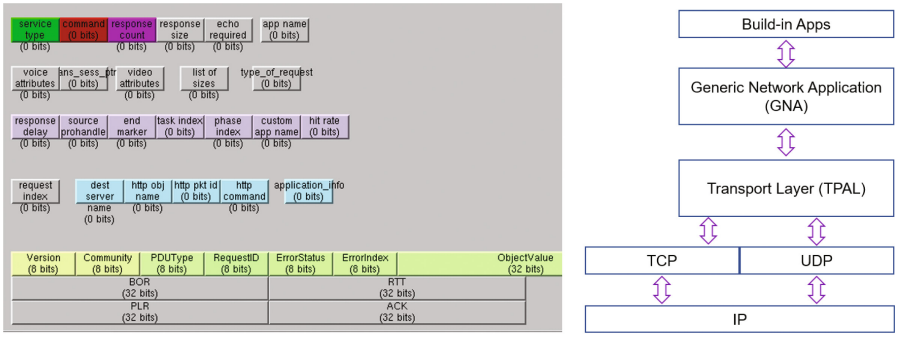
3.1 SNMP Agent Implementation

We created a new process model supported by *gna-clsvr-mgr4-Router* that was based on the application model of OPNET Modeler to support SNMP. This process model is able to process both the regular packets and the *getRequest* and *getResponse* packets. By embedding SNMP messages inside the Generic Network Application (GNA) packet, it can also exchange SNMP messages between the client and server at the application layer by sending GNA packets.

Fig. 3. SNMP message exchange mechanism and architecture.

The SNMP message format was created for exchanging messages between the agent manager and the intelligent routers. The SNMP *getRequest* and *getResponse* messages, embedded in the GNA packet, can be exchanged between the agent manager and the intelligent routers. Figure 3 shows the design of the SNMP message format. There are 11 columns in the message format, and they support SNMP version 2. The agent manager can exchange information with the intelligent routers using the SNMP message format.

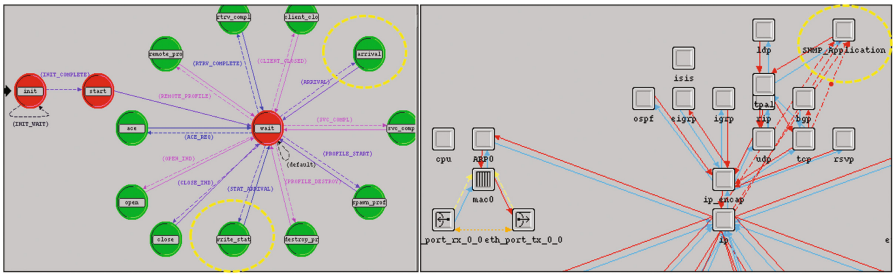
The clients and server in this application use the GNA packet format to exchange messages at the application layer of OPNET Modeler. In order to exchange SNMP message information, the SNMP message was embedded in the data field of the GNA messages, which are supported in the OPNET Modeler. Figure 4 shows the SNMP message embedded in the GNA packet of OPNET Modeler.



(a) GNA packet format of OPNET Modeler (b) Build-in Applications in OPNET

Fig. 4. GNA packet format of OPNET Modeler.

The *gna-clsvr-mgr4-Router* is depicted in Fig. 5(a). This process model parses *getRequest* packets in the Arrival process and sends *getResponse* packets to the agent manager by invoking the *gna-clsvr-mgr-getRespond-packet-send* function. The write-stat process is used to record the current operational data of a router. In order to support the SNMP application for a router, this SNMP-Application process model was added to the process model architecture of a router and connected to the Transport Layer (TPAL) process model with two stream wires. As shown in Fig. 5(b), four statistic wires were created to collect the operational data of a router, such as BOR, RTT, PLR, and ACK.



(a) Process model of *gna_clsvr_mgr_4_Router* (b) Model of the intelligent router

Fig. 5. Process and model for i-SNMP integrated in the Router.

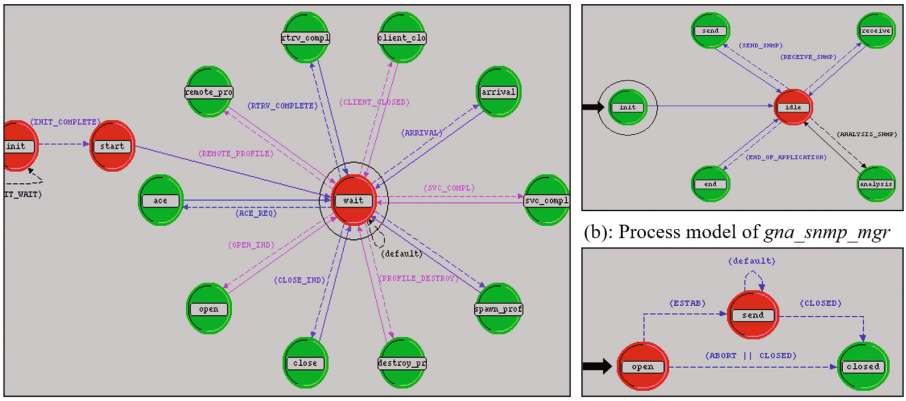
3.2 SNMP Agent Manager Implementation

In this session, three new process models were created: *gna-clsvr-mgr4-MgrSNMP*, *gna-snmp-mgr*, and *gna-snmp-cli* for the agent manager to support the SNMP application. These three process models deal with sending and receiving regular packets as well as SNMP *getRequest* and *getResponse* packets between a client and server at the application layer, saving information on intelligent routers, and analyzing the real-time status of the intelligent routers. The *gna-clsvr-mgr4-MgrSNMP* is depicted in Fig. 6(a). This process model of the agent manager is used for processing the GNA packets at the application layer. This process model is modified to support the SNMP application based on the process model *gna-clsvr-mgr* of OPNET Modeler. Figure 6(b) shows the complete design of the *gna-snmp-mgr* process model. This process model has four functions: send, receive, analysis, and end.

- In the send and receive functions, the sub-process *gna-snmp-cli* is invoked to send *getRequest* packets and receive *getResponse* packets every 10 s.
- The analysis function retrieves collected data from its database every 30 s, analyzes the current status of intelligent routers, and detects the three network faults of intelligent routers.
- The end function closes the session between the client and server after it receives a packet with a close command. The design of the *gna-snmp-cli* process model is shown in Fig. 6(c).

This process model has three functions: open, send, and closed. It helps the process model of *gna-snmp-mgr* carry out the task of sending and receiving packets.

- The open function sends *getRequest* packets to the intelligent routers.
- The send function is invoked to receive the *getResponse* packets from the intelligent routers and save the information to the database of the agent manager.
- At last, the closed function closes the session between the client and server after it receives a packet that contains a close command.



(a): Process model of *gna_clsvr_mgr_4_MgrSNMP*

(b): Process model of *gna_snmp_mgr*

(c): Process model of *gna_snmp_cli*

Fig. 6. Process model of SNMP Agent Manager.

Figure 7 illustrates the database table of the agent manager. There are nine columns that keep the information of intelligent routers, such as "Version", "Community", "PDUType", "RequestID", "TimeStamp", "BOR", "RTT", "PLR" and "ACK". The agent manager retrieves the information from its database, analyzes the real-time status of the intelligent routers, and detects the four network faults of the intelligent routers.

Version	Community	PDUType	ReqestID	TimeStamp	BOR	RTT	PLR	ACK

Fig. 7. Database table of the agent manager.

The network environment setup focuses on showing the ability of the agent manager to monitor the status of a group of distributed intelligent routers by using SNMP. In this paper, the network topology was designed to be a single network community with three subnets. Three intelligent routers connect three subnets using 10Mbps Ethernet connections. Figure 8 illustrates that three intelligent routers are distributed in three different subnets. In Subnet-0, the agent manager monitors the three intelligent routers and maintains a database that keeps the information collected from all three intelligent routers.

In order to simulate the regular network traffic, an application server and an HTTP server were set up to provide the network clients with applications such as HTTP and databases. The network node "Application Configuration" of OPNET Modeler is used for configuring the applications on the server side, and "Profile Configuration" is used for configuring the profiles on the client side.

The network topology is depicted in Fig. 9(a). There are three distributed subnets and three configuration nodes. Subnet0's topology is depicted in Fig. 9(b). There are five network devices in this subnet. The agent manager

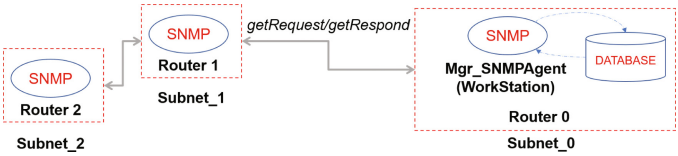
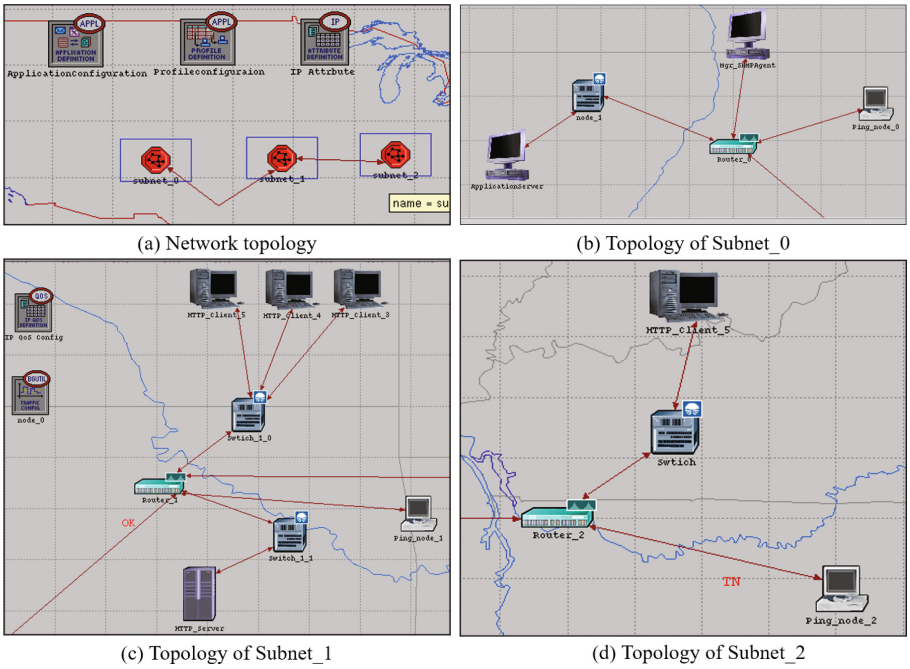


Fig. 8. Distributed network routers and agent manager.



(a) Network topology

(b) Topology of Subnet_0

(c) Topology of Subnet_1

(d) Topology of Subnet_2

Fig. 9. Network System in OPNET Modeler.

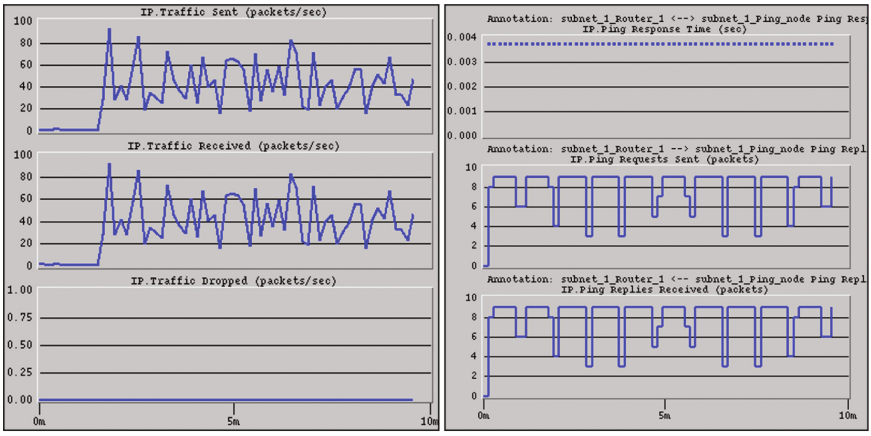
is located in this subnet. Subnet1's topology is depicted in Fig. 9(c). There are eight network devices and two network configuration nodes in this subnet. The simulation of network faults is configured by the "IP QoS Config" and the "Traffic Config". Subnet2's topology is depicted in Fig. 9(d). There are four network devices in this subnet.

4 Scenario Simulations and Results

In this session, four scenarios are simulated to test and demonstrate the SNMP intelligent agent's ability to detect network faults. The SNMP agent manager sends out requests to get information from network nodes and gets back SNMP messages that contain the data. The data are then written to the database, which is a generic data file. An Analysis process is added to perform the analysis task.

The analysis process analyzes the collection of data for RTT, acknowledge, BOR, and PLR; diagnoses the status of the router; sets the router information; writes the router information to the database; and returns a pointer to the router information with the router's most current status.

The router information can be used by other applications that need to know the current status of network nodes. To discover the status of the router, the analysis process first compares the data with the lower limits to see if there is any sign of a problem. If PLR, BOR, and RTT are smaller than the lower limit, there is no sign that a problem occurred. If any of these are above the limit, the analysis process would look at the slope of the ten most recent records and calculate the slope of the regression line, with the horizontal data being the time stamp of the SNMP message and the vertical data being the data, such as BOR. If the slope of the regression line is less than the threshold of slope, the analysis process would consider that a normal condition. Otherwise, further investigation is needed to determine if there is a network fault. We will discuss the different patterns that appear on the operational data when network faults occur in the following.



(a) Results of Packet Loss (b) Results of Round-Trip Time

Fig. 10. Packet Loss and RTT results for Normal Traffic.

4.1 Scenario1: Regular Traffic

Scenario Description: In subnet-1, a few HTTP-client nodes are added. A ping-node is added and connected to Router-1. HTTP-client workstations that run the HTTP (heavy) profile, which accesses the HTTP server in the same network as HTTP client/server applications, are analogous to how workstations and servers run applications in real life. Between Router-1 and Ping-node, the

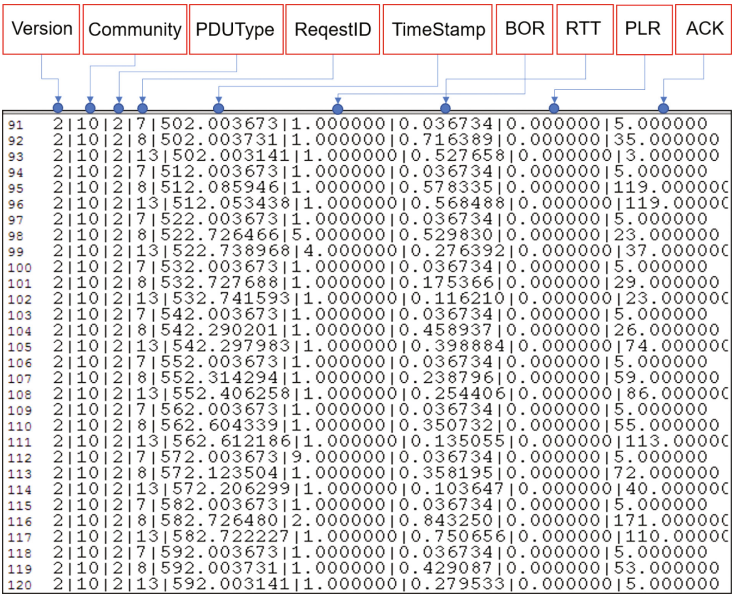


Fig. 11. Node Information for Normal Traffic.

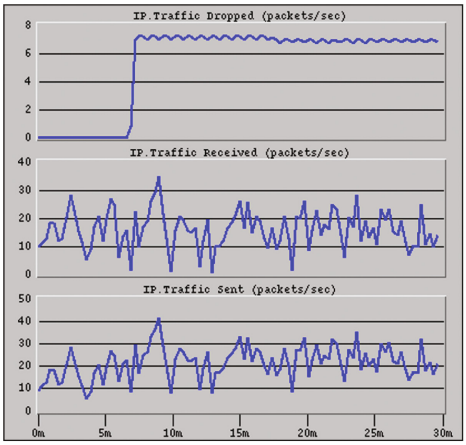
PING pattern uses *PacketSize* 1024. This pattern will not cause network congestion because of its small packet size and long timeout. Inter-repetition time is set to 7.0 s, meaning a PING packet is sent out every 7 s. The maximum repetition count is unlimited, meaning PING traffic will last as long as the simulation runs. Subnet-2 is similarly configured. The HTTP clients in Subnet-2 access the HTTP server in Subnet-1, and PING traffic in Router-2 is set up the same way as in Router-1.

Scenario Results: As shown in Fig. 10(a), the simulation result indicates that IP traffic on Router-1 is stable and does not vary dramatically. Packet loss on Router-1 is zero. There is no congestion; all IP packets are delivered, and replies are received. In Fig. 10(b), this result showed that the PING response time, which is the RTT, is constant. It was also noted that the PING requests sent, and PING replies received were almost identical, which meant the traffic was normal and no congestion occurred. For this reason, there is no increase in the buffer occupancy of the FIFO queue.

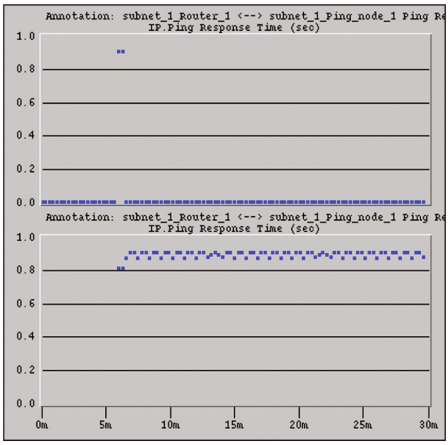
Analysis Result: In the database of the SNMP agent manager, as shown in Fig. 11, the BOR is less than 10, the RTT is less than 1, the PLR is 0, and the ACK is mostly constant. The analysis showed that all three routers are in normal condition.

4.2 Scenario2: Highly Congestion

Scenario Description: HTTP clients and the server still run the same traffic as in a regular traffic scenario, but the setting for PING traffic on Router-1 is different. In the first six minutes (360 s), it runs the same light PING traffic as in regular traffic. It runs heavy PING traffic after 360 s. The very-big-packet-size PING pattern is used. The packet size in this pattern is very large (10000), and the timeout is only 2.0 s. With large packets like this, it's very likely to cause congestion in the network. The processing speed of the router on the interface, which generates and receives the PING traffic, is also slowed down to exaggerate the congestion. Running regular traffic in the first six minutes gives a comparison between regular traffic and congestion traffic.



(a) Results of Packet Loss



(b) Results of Round-Trip Time

```
51  2|10|2|13|369.004191|0.000000|0.020956|0.000000|1.000000
52  2|10|2|7|378.477438|2.000000|0.000000|0.000000|4.000000
53  2|10|2|8|378.000400|13.400000|714.839355|0.000000|25.000000
54  2|10|2|13|379.004191|0.000000|0.020956|0.000000|1.000000
55  2|10|2|7|388.477438|2.000000|0.000000|0.000000|4.000000
56  2|10|2|8|388.000400|20.600000|1061.489258|0.000000|25.000000
57  2|10|2|13|389.004191|0.000000|0.020956|0.000000|2.000000
58  2|10|2|7|398.477438|2.000000|0.000000|0.000000|21.000000
59  2|10|2|8|398.000200|27.400000|1274.288086|0.000000|21.000000
60  2|10|2|13|399.004191|0.000000|0.020956|0.000000|2.000000
61  2|10|2|7|408.477438|9.000000|0.000000|0.000000|22.000000
62  2|10|2|8|411.000400|36.300000|1827.281250|0.000000|8.000000
63  2|10|2|13|412.004191|0.000000|0.020956|0.000000|5.000000
64  2|10|2|7|418.477438|2.000000|0.000000|0.000000|4.000000
65  2|10|2|8|418.000400|41.300000|961.727051|0.000000|24.000000
66  2|10|2|13|419.004191|0.000000|0.020956|0.000000|2.000000
67  2|10|2|7|428.477438|2.000000|0.000000|0.000000|17.000000
68  2|10|2|8|428.000400|48.400000|961.727051|0.000000|24.000000
69  2|10|2|13|429.004191|0.000000|0.020956|0.000000|2.000000
70  2|10|2|7|438.477438|2.000000|0.000000|0.000000|4.000000
71  2|10|2|8|438.000400|50.000000|2644.748047|35.000000|70.000000
72  2|10|2|13|437.004191|0.000000|0.020956|0.000000|5.000000
73  2|10|2|7|448.477438|2.000000|0.000000|0.000000|4.000000
74  2|10|2|8|448.000400|50.000000|2740.921875|36.000000|68.000000
75  2|10|2|13|449.004191|0.000000|0.020956|0.000000|3.000000
76  2|10|2|7|458.477438|2.000000|0.000000|0.000000|4.000000
```

(c) Results of Node Information

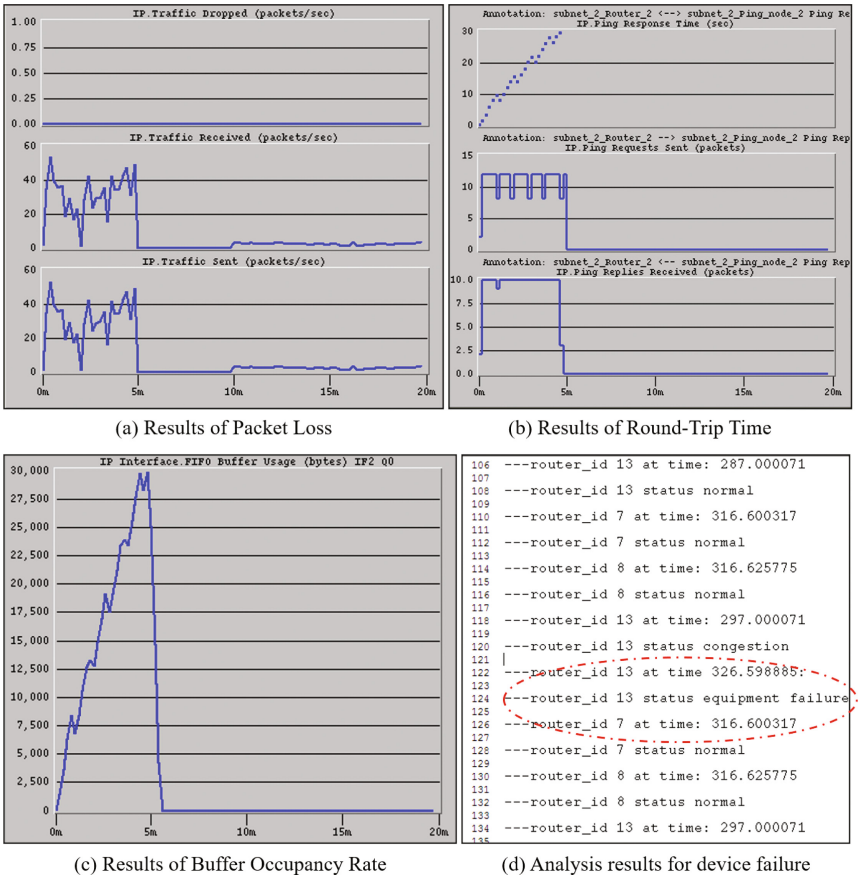
```
19185 node-8
19186 timestamp-398.000200
19187 BOR-27.400000
19188 RTT-1274.288086
19189 PLR-0.000000
19190 ACK-21.000000
19191 ---router status normal
19192
19193 node-8
19194 timestamp-411.000400
19195 BOR-36.300000
19196 RTT-1827.281250
19197 PLR-0.000000
19198 ACK-8.000000
19199 ---router status congestion
19200
19201 node-8
19202 timestamp-418.000400
19203 BOR-41.300000
19204 RTT-961.727051
19205 PLR-0.000000
19206 ACK-24.000000
19207 ---router status congestion
19208
19209 node-8
19210 timestamp-428.000400
19211 BOR-48.400000
19212 RTT-961.727051
19213 PLR-0.000000
```

(d) Analysis Results

Fig. 12. PLR and RTT results for highly congestion traffic.

Scenario Results: Fig. 12(a) shows that after six minutes (360 s), the IP traffic dropped, which is the packet lost, increases dramatically. Because the processing speed cannot match the speed at which packets are linked to this interface, the BOR increases from 360 s. Packets have to remain in queue for a longer time. Those packets with short timeout limits will be lost. As a result, the RTT after 6 min has increased significantly, as illustrated in Fig. 12(b).

Analysis Result: In this scenario, the slopes of PLR, BOR, and RTT are bigger than the thresholds, and ACK is constant or decreasing. The analysis process determines that there is traffic congestion. In another case, if PLR, BOR, and RTT are bigger than the upper limits, it's identified as a congestion condition because the slope at a certain point will flatten and will not increase significantly anymore. Figures 12(c) and (b) show that the BOR and PLR of Router-1 (node ID: 8) are increasing, indicating traffic congestion, based on node information.



(a) Results of Packet Loss

(b) Results of Round-Trip Time

(c) Results of Buffer Occupancy Rate

(d) Analysis results for device failure

Fig. 13. PLR and RTT results for Node Failure.

4.3 Scenario3: Node Failure

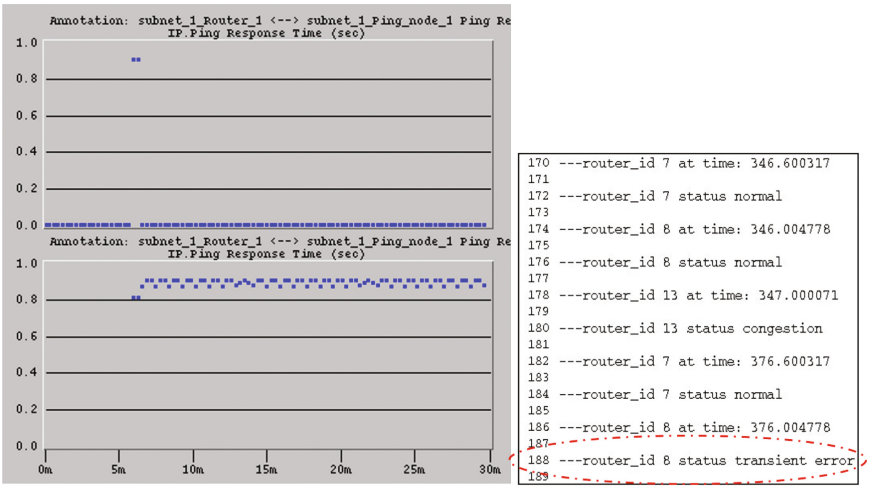
Scenario Description: The device node failure scenario is set up in the following way: on Router-2, node failure is simulated in subnet-2. Subnet2's traffic setup is the same as that in Scenario 1. To configure a node failure, a Failure/Recovery config node is added. The attribute is specified in the node failure and recovery specification. A node failure and recovery have to be entered in pairs. It specifies the start time of failure and the start time of recovery for a particular node. The time interval in between is the duration of the failure. In this scenario, Subnet-2 and Router-2 will fail for five minutes, which is from 300 s to 600 s.

Scenario Results: As shown in Fig. 13(a), the result indicates that IP traffic sent and received drop to zero after five minutes in this simulation; packet loss drops to zero because the device is down and then comes up a little after the 5 min down time. As shown in Fig. 13(b), the RTT increases in the first five minutes. Because Router-2 is down, the RTT drops to zero after five minutes. The BOR increases in the first five minutes, as shown in Figs. 13(c) and (d). Then, it drops to zero after five minutes because this device is down. According to the analysis results, Router-2 (node ID 13) has a device failure after five minutes.

4.4 Scenario4: Transient Error

Scenario Description: A timestamp is set up for the start point of any error that occurs. If the problem persists longer than the time frame for a transient error, it is identified as a persistent error. Otherwise, it is identified as a transient error. Subnet-1 simulates a link failure for the link Ping-node-1 and Router-1. Traffic is configured the same as in Scenario Regular Traffic. In order to simulate link failure, a Failure/Recovery Config node is added. Failure and recovery are added in pairs. The time when failure occurs at the link is specified as 360 s. The time when the link will be recovered is specified as 380 s. The duration of the failure will be 20 s. From 360 s to 380 s, the link between Ping-node-1 and Router-1 in subnet-2 fails.

Scenario Results: As shown in Fig. 14, the RTT is high only for a short period of time after six minutes when the link is broken. And the analysis result identifies the transient error at Router-1 (the node ID is 8) after six minutes.



(a) Results of Round-Trip Time (b) Analysis Results of Transient Error

Fig. 14. RTT and analysis results for Transient Error.

5 Conclusions

Various traffic combinations can be configured in OPNET Modeler to simulate different scenarios in the network environment. Results show that our proposed i-SNMP agent and the SNMP agent manager are able to collect these simulation statistics, perform analysis, and pinpoint the network faults. The SNMP process is implemented in a way that is easy to configure, automatically starts, and requires little maintenance. It can be added to all kinds of network nodes that support TCP/IP and client/server application processes. Both the SNMP agent and SNMP manager processes can be easily implemented and configured in future research.

References

1. Cisco, Network Management System: Best Practices White Paper, Technology White Paper, August 2018
2. Roquero, P., Aracil, J.: On performance and scalability of cost-effective SNMP managers for large-scale polling. IEEE Access **9**, 7374–7383 (2021)
3. Pallmann, D.: Network query language (NQL). Wiley, New York (2002)
4. Safrianti, E., Sari, L.O., Sari, N.A.: Real-time network device monitoring system with simple network management protocol (SNMP) Model. In: 2021 3rd International Conference on Research and Academic Community Services (ICRACOS), IEEE (2021)
5. Helali, S.: Monitoring systems and networks. In: Systems and Network Infrastructure Integration: Design, Implementation, Safety and Supervision, Wiley, pp. 157–171 (2020)

6. Aweya, J.: Designing Switch/Routers: Architectures and Applications. CRC Press, Boca Raton (2022)
7. OPNET Technologies Inc, Opnet Modeler. http://www.opnet.com
8. Smera, C., Sandeep, J.: Networks Simulation: research based implementation using tools and approaches. In: 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT). Bangalore, India, pp. 1–7 (2022)
9. Yang, B.: Research on simulation system of information management construction based on computer internet technology. In: Proceedings of the 7th International Conference on Cyber Security and Information Engineering (2022)
10. Chen, M., Miao, Y., Humar, I.: OPNET IoT Simulation. Springer, Singapore (2019). https://doi.org/10.1007/978-981-32-9170-6
11. Chen, M.: OPNET Network Simulation, Press of Tsinghua University, ISBN 7-302-08232-4 (2004)