Design and Implementation of a Lightweight Application for Network Overload Monitoring

Marija Milosevic, Milica Spasic, Vladimir Ciric

***Abstract*—The ever-growing use of technology has led to network systems becoming an essential component of modern organizations, whose functioning highly depends on the ability to transfer and access data in near real-time. Having that in mind, it is important to keep the network efficiency, availability and reliability at a high level. Analyzing computer network traffic holds a key role in today's complex networking systems, as potential network overload can lead to delayed or lost network packets, or even disruption of upcoming connections. To improve network availability and performance, as well as to detect problems, it is very important to have insight into where and how data flows within the network. The goal of this paper is to design and implement a lightweight application for agentless network overload monitoring that would help alleviate the difficulties network administrators are facing while maintaining the network. The proposed design visualizes network traffic paths and creates a network baseline as a reference point for further comparison. A use-case of the proposed solution is given.**

***Index Terms*—Computer networks; Network monitoring; Network baseline; Data visualization**

I. Introduction

The computer industry has progressed rapidly over the past few decades. With new components and systems developing constantly, as well as new services being introduced every day, there is a vast amount of expectations regarding connectivity, availability and networking that need to be met [1]. A computer network is a group of two or more devices that can communicate with each other and share resources, and are connected by an adequate medium and intermediary devices. The availability and flexibility of modern computer networks allows anyone, from anywhere, to connect to the network and access the desired information. Nowadays, real-time data transmission (voice, video, etc.) is often a requirement [2]. Computer networks must provide general, cost-effective, fair, and robust connectivity among a large number of computers. Along with this, networks must constantly evolve to accommodate changes in both the underlying technologies on which they are based, as well as the demands placed on them by application programs. Creating a network that meets these requirements is not a simple task [3].

Marija Milosevic is with the Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia (e-mail: marija.milosevic@elfak.ni.ac.rs), (https://orcid.org/0000-0002-3658-0292)

Milica Spasic is with the Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia (e-mail: milica.spasic@elfak.rs)

Vladimir Ciric is with the Faculty of Electronic Engineering, University of Nis, Aleksandra Medvedeva 14, 18000 Nis, Serbia (e-mail: vladimir.ciric@elfak.ni.ac.rs), (https://orcid.org/0000-0002-1442-7959)

Network availability and speed are crucial for developing a successful business, as well as for more efficient use of available resources. If large amounts of data are transmitted through the network, it can have a negative impact on the efficiency of the system. Also, malicious groups or individuals can cause damage in many ways. In recent times, more and more organizations rely on machine learning and deep learning implementations for inspecting and visualizing the network, along with detecting and preventing potentially malicious traffic [4].

Computer network congestion most often occurs when data that reaches a network node exceeds its processing capabilities, which results in reduced network service capacities. In a network as a whole, this leads to delays in receiving data, packet loss or blocking new connections [5]. To be able to identify potential problems on the network, there needs to be a sense of what is normal and expected network behaviour. For defining expected behaviour, network baselining needs to be implemented over a defined time period. This process assists a network administrator determine the thresholds at which an alert has to be triggered, and helps maintain network stability and reliability, along with achieving maximum network performance, all while preventing network downtime [6].

Monitoring network traffic facilitates more accurate capacity planning and ensures that resources are used in the best way possible [5]. Graphical representation and visualization provide a better insight into communication on the network. This can be of great help for detecting and repairing weak points in the network, detecting unauthorized traffic, security analysis, etc.

In this paper we observe the operation of the network and propose an application for visualization of network load. The proposed design is an agentless solution primarily based on ICMP protocol, which monitors network traffic, visualizes traffic paths and creates network snapshots as a reference point for further comparison. To the best of our knowledge, no available software solution provides saving and displaying the network baseline nor comparing current network paths with the obtained baseline.

The paper is organized as follows: Section 2 gives a theoretical background regarding computer networks and monitoring. In Section 3 , the proposed design of the application and its implementation is presented. In Section 4, we give a demonstration of the application operation regarding the network of Faculty of Electronic Engineering in Niš. Concluding remarks are given in Section 5.

 ISBN 978-86-7466-970-9

II. Background

A. Network Monitoring

Network monitoring allows network administrators to get a better insight into what is happening in the network, regardless of the network type and topology, as well as the state of various network nodes, which include access, distribution and core switches, firewalls, routers, servers, client systems, etc. Knowledge of networking components aids in network management and monitoring.

Regardless of the software, protocol, or technique used, a few fundamental aspects of a monitoring system are universal. A single aspect of a monitored device can return at least one piece of information. Along with this, acquisition and frequency are terms used to describe how and how often the information is gathered. Another important aspect is the threshold, which is, in its simplest form, the chosen baseline value for a certain network parameter. When that value is beyond the threshold, an alert is triggered [7].

Network monitoring tools can be agent-based, agentless or, though rarely, a hybrid of the two. An agent is a software that is installed on every monitored device and has access to the device's performance data. The most common method for an agent-based monitoring system is to send data at predefined intervals. While this type of monitoring allows for more granular data to be obtained, it is time consuming and requires maintenance. Agentless monitoring, however, uses remote APIs that are exposed by the service that needs to be monitored, or analyzes data packets being transferred to and from the monitored device. It gives a more lightweight approach to network monitoring by eliminating the need to deploy or maintain agents throughout the network [7].

There are various techniques for implementing network monitoring and some of the most utilized are:

- Ping - This is a utility used to test the availability of a host in an IP network. From the obtained ping results it can be determined whether a host in the network is active or not. It is based on the ICMP protocol.
- Simple Network Managment Protocol (SNMP) - SNMP is used to identify network devices, monitor network performance, keep track of network changes and determine the status of network devices in real time [8].
- Syslog - Syslog is a message logging system that allows a variety of devices in IP networks to send event notifications. The information contained in these messages can be used for both system management and security auditing [7].
- Netflow - This is a service which provides network administrators access to IP flow data from networks. Flow data is collected by network devices (routers and switches) and exported to collectors. The collected data allows for fine-grained metering, which can result in highly flexible and detailed resource usage metrics [9].

Various tools have been created for network monitoring, some of which are open-source, while others offer trial versions. Tools like Open Visual Traceroute, or Cisco's ThousandEyes offer a wide range of monitoring options. However, none of the tools mentioned above don't have an option for saving the obtained baseline for further comparison with new data.

B. ICMP Protocol

The Internet Control Message Protocol (ICMP) is a network layer protocol used to check the status of a network or a specific link and provide feedback about communication problems [10]. ICMP messages are used when a datagram cannot reach its destination or when the gateway lacks the buffering capacity to forward a datagram. ICMP messages are sent using a basic IP header. Some of the most common ICMP message types are *Destination Unreachable*, *Time Exceeded*, *Parameter problem*, *Source quench*, *Redirect*, *Echo/Echo reply*, *Timestamp request/Timestamp reply* [11].

Ping, traceroute, pathping, and other network management tools utilize ICMP. Fig. 1 shows how ping operates in order to test the availability of the device - it sends an ICMP echo request to a system and waits for the target to send back an ICMP echo reply message. With the use of ping, we can determine whether the remote system is online and responding promptly, as well as the speed at which packets travel from one host to another [12].

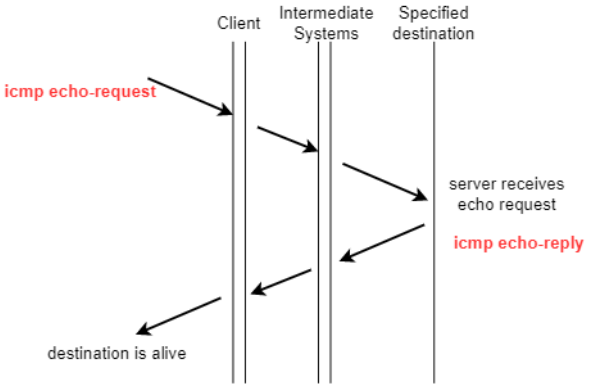

Fig. 1. Use of ping to test host availability.

Traceroute is a utility designed to determine the path taken by a data packet as it travels from its origin to its destination. While ping gives a response if the targeted host is up and running, traceroute sends a response with each passed router or hop on the way to the target address, modifying the TTL field in the IP header along the way [12]. Both ping and traceroute can give, among other information, the time between initiating a network request and receiving a response - network round-trip time (RTT) - measured in miliseconds. Routers can be configured not to process ICMP messages, or not to respond to them. In that case, routers are hiding the network topology, and no RTT will be obtained for that segment [3].

III. Design and Implementation

The application proposed in this paper is a simple solution that could help alleviate the difficulties network administrators face while maintaining the network. It provides tracking

 ISBN 978-86-7466-970-9

response time of multiple hosts, its visual representation and statistics, network topology representation, saving and displaying the network baseline and comparing new data with the previously established baseline. It uses simple utilities - Ping and Traceroute, which have been mentioned in the previous section.

The proposed application is developed in C#, using the MVC (Model-View-Controller) software architectural pattern. This pattern separates internal data representation and business logic from how information is presented to and accepted by the user. Fig. 2 presents the system architecture, with all MVC components, along with an additional Commands component, which implements the main funcionalities such as ping and traceroute, and uses the *System.Net.NetworkInformation* namespace.

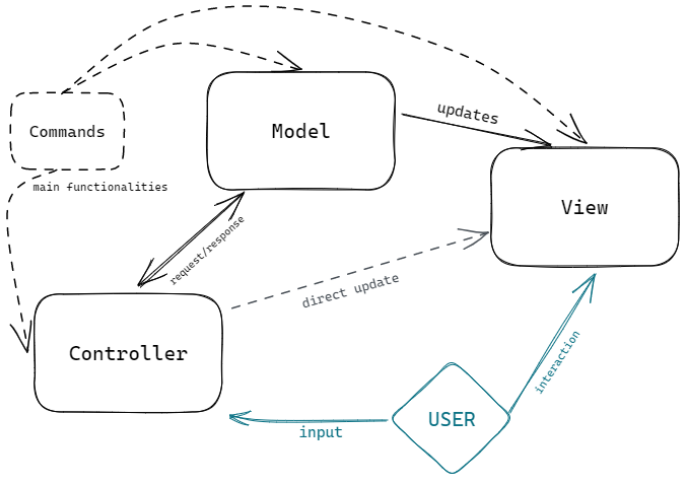

Fig. 2. System architecture.

Three different use cases were identified during design: Data flow, Graph and General, and for each of them the appropriate Controllers and Views are implemented. Data flow includes tracking response time of multiple hosts using ping as a utility. It is represented to the user via a chart ASP.NET form. Graph use case refers to drawing traffic paths to multiple hosts, their saving, displaying, and manipulation. For generating and presenting graphs, an open source tool, MSAGL, is used. It gives the option to manipulate the graph view, as well as save the graph in bitmap or vector format. Third use case, General, simply executes ping and traceroute over one host and gives insight into the structure of the data used in the two more complicated use cases.

A. Functionality

The implemented application gives various options for monitoring the network. In order to provide multiple host monitoring, options for adding, modifying and deleting hosts are implemented. A host can be defined by its IP address or domain name. For achieving flexibility in ping and traceroute settings, an option to set the frequency of gathering host response times has been added. Gathering response times can be done every X seconds, or by sending Y ping requests at once. Furthermore, a maximum number of hops for traceroute and the number of ping requests per hop can be modified. One of the most important options is defining the threshold, a value that indicates a drastic decline in network performance. The user can set an acceptable threshold and an alarming threshold, where the former gives a tolerable deviation from mean RTT, while the latter defines an unacceptable difference. Depending on the obtained results and its comparison with the threshold, the information depicted in a graph will have a different colour. The option to change the threshold provides the user with a flexible solution where he can decide what behaviour is tolerable on the network.

The data flow form can be seen in Fig. 3. For the observed hosts, multiple pings are performed and information about the RTT of last ping, average success RTT, number of lost packets and a total number of sent packets are given. On the right side of the form is a chart representing the RTTs for the last N ping requests sent to every host. This chart is updated as new ping information arrives.

The Graph form utilizes traceroute and depicts a graph consisting of the paths from the origin device to all selected host devices. Each node in a graph is a contacted device on the way to some of the selected hosts. The numbers on the edges present obtained RTT between node A and node B expressed in milliseconds. Fig. 4 depicts a graph for several contacted hosts. A path to every host has been displayed with edges of different color. From Fig. 4 it can be seen that there are multiple edges from localhost (127.0.0.1) to 160.99.13.11 - all the hosts have this node in their path. It can be observed that some nodes are contacted almost instantly, as their RTT is 0 ms. Moreover, a couple of nodes along the path to one host (*www.netacad.com*) are contacted with the RTT of '*', and the IP address is unknown. These nodes try to hide the network topology. Along with this, current graph information can be saved as a baseline for further comparison. The information is kept in JSON format, where files are created for every host, and every edge has a timestamp, average RTT, a hostname and an address. Comparison can be made based on the time difference in the saved baseline or a time tag, which can be useful for monitoring recent events.

IV. A Use-case of the Proposed Solution

In this section, the application functionality has been demonstrated through monitoring the network of Faculty of Electronic Engineering, University of Niš, Serbia.

The hosts chosen for monitoring are essential nodes for the network, but also the faculty as an organization. We included three main faculty websites, a mail server, two DNS servers, the university's website, as well as one of the hosts located in the Laboratory for Advanced Cybersecurity. They have all been added with their domain names, but could have also been referenced with their respective IP addresses.

Fig. 3 gives the entire data flow view for the observed hosts. The left part of the view gives the last RTT and average success RTT values for each host, as well as the number of

 ISBN 978-86-7466-970-9

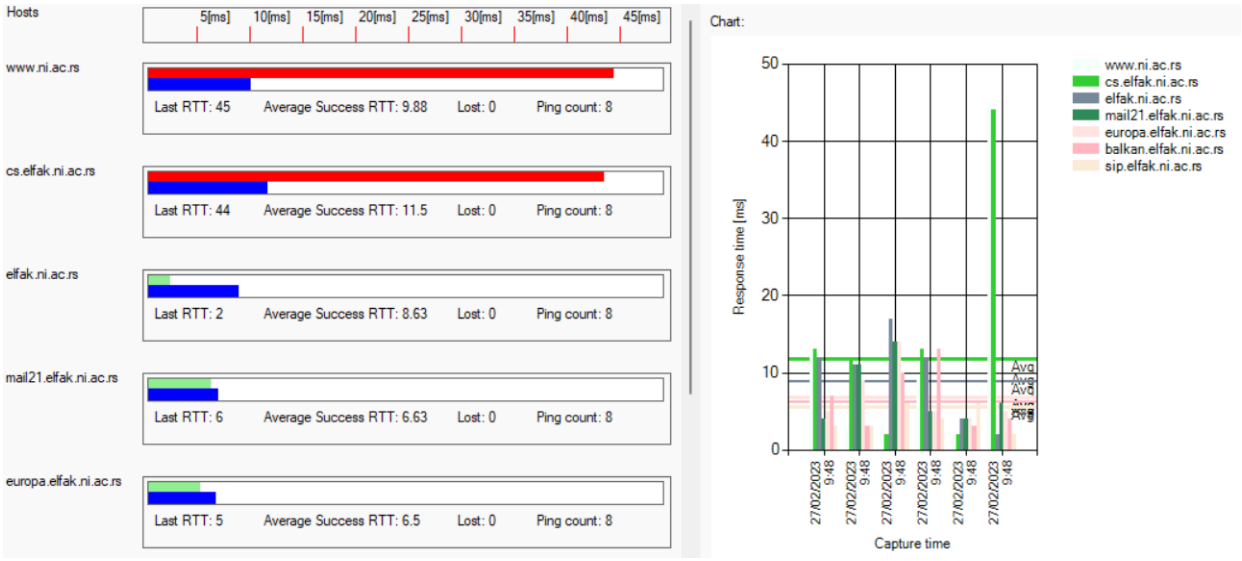

Fig. 3. Data flow view for local network hosts.

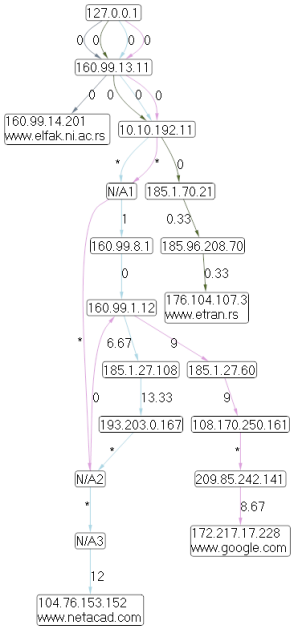

Fig. 4. A graph view for hosts in remote networks.

lost packets and ping requests. The chart on the right side of the form makes deviations easier to detect. Two of the hosts (www.ni.ac.rs and cs.elfak.ni.ac.rs) have had a higher RTT at the last ping iteration, which can be noticed as a peak in the rightmost values in the chart. This RTT will affect the average RTT time at the next ping iteration.

A graph consisting of paths to the selected hosts, as well as the comparison to the baseline, is shown in Fig. 5. Before this graph was generated, at least one traceroute has been issued by the user to all hosts and a baseline was created. If, when saving the baseline, contacting the host was not made by the same route as later, some edges will not have a baseline value for comparison. This can be seen with two edges on the path to the *www.ni.ac.rs* host. Additionally, apart from the RTT on the edges depicted in Fig. 4, another information is present in the baseline comparison from Fig. 5: RTT from the baseline (starting with the letter 'B'), expressed in milliseconds. The chosen acceptable and alarming thresholds are 100% and 200%, respectively. In the previous section we mentioned that, depending on the baseline comparison and the given threshold values, the information presented in the graph will be coloured differently. In Fig. 5 a distinction can be made between edges whose new RTT values are over the defined alarming threshold compared with the baseline values, which are marked red, and those that are somewhere between the thresholds or under the acceptable threshold, marked orange and green. This visual distinction helps with detecting a potential problem. From Fig. 5, it can be noticed that there has been some sort of delay on the network, as almost all nodes have gone over the alarming threshold. A couple of things need to be taken into consideration: even though almost all RTT values are higher than their baseline pairs, these values are of the order of 15ms. Depending on the network and its requirements, this amount of time can be considered acceptable, tolerable or alarming. This is the main reason why threshold is an important parameter that needs to be calibrated through network monitoring periods.

V. Conclusion

The concerns for network administrators are growing bigger each day, as organizations broaden their networks and improve their systems. In this paper, a lightweight application has been designed and implemented for the purpose of network load monitoring. The application relies on ICMP protocol and basic utilities such as ping and traceroute. It can be used without

 ISBN 978-86-7466-970-9

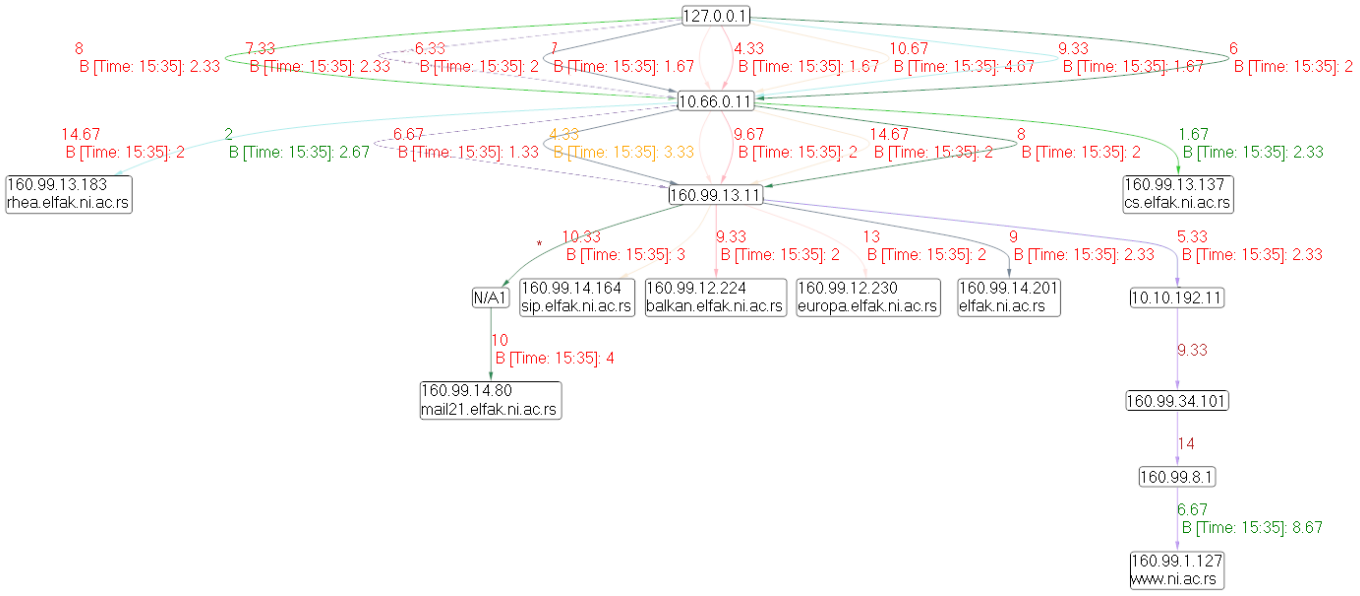

Fig. 5. Baseline comparison for the chosen local network hosts.

any additional device settings and does not need any agent software to function properly. Since the proposed solution relies on RTT values, there can be difficulty in establishing whether the problem is on the egress or on the ingress path - more advanced methods need to be used in order to obtain that information. This solution provides efficient visualizations for multiple hosts and offers significant flexibility to the user, providing options for thresholds, ping count and frequency, baseline saving, displaying and comparing. The application use-case has also been presented on the example of the faculty's network.

ACKNOWLEDGMENT

This work was supported by the Serbian Ministry of Education, Science and Technological Development [grant number 451- 03-68/2022-14/ 200102].

REFERENCES

[1] Beqiri E. , "The Implications of Information Networking at the Pace of Business Development", Knowledge - International Journal, Vol. 27, November 2018.
[2] A.S.Tanenbaum, Computer Networks, 5th Edition, 2011.
[3] Peterson, Larry L. and Davie, Bruce S., "Computer networks : a systems approach", 5th ed.(The Morgan Kaufmann series in networking), 2012. ISBN: 978-0-12-385059-1.
[4] Milosevic, M. S., & Ciric, V. M., "Extreme minority class detection in imbalanced data for network intrusion", Computers & Security, Vol. 123, 102940, December 2022. https://doi.org/10.1016/j.cose.2022.102940
[5] Welzl M. , "Network Congestion Control", John Wiley & Sons, 2005.
[6] Nassar D. , "Network Performance Baselining", MTP, 2000.
[7] Adato, L, "Monitoring 101: A primer to the philosophy, theory, and fundamental concepts involved in systems monitoring", Solarwinds, 2019.
[8] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol (SNMP)", RFC 1157, DOI 10.17487/RFC1157, May 1990, https://www.rfc-editor.org/info/rfc1157.
[9] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, DOI 10.17487/RFC3954, October 2004, https://www.rfc-editor.org/info/rfc3954.
[10] Chapple, M , Stewart, J. M., & Gibson, D. "Certified Information Systems Security Professional", 8th edition, John Wiley & Sons, 2018. In-Text Citation: (Stewart et al., 2005)
[11] Postel J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, https://www.rfc-editor.org/info/rfc792.
[12] Gregg, M., "The Network Security Test Lab: A Step-by-Step Guide", John Wiley & Sons, 2015. ISBN: 978-1-118-98705-6