Analyzing of Critical Points of Corporate Network for Upgrading Its Scalability

N. Dimov, D. Nyagolov, R. Krystev

Abstract **— Networking technologies have evolved significantly over the years as Ethernet and Wi-Fi requirements have increased significantly. In addition to supporting a range of devices, LANs require managing traffic generated from many other sources such as live streaming video, video surveillance, network attached storage (NAS), voice over IP (VoIP), virtualization, cloud and IoT devices and services generate demand for additional bandwidth and low latency. It is clear that legacy network equipment and cabling is not ready to handle the increased data traffic, creating the need for research and analysis to build an efficient and low-cost solution. In this study, the main problems encountered in the specific enterprise LAN are discussed and analyzed.**

Keywords **— network, bandwidth, latency, performance**

I. INTRODUCTION

In today's technetronic world, where people are immersed in a vast ocean of information [3], the topic of connectivity is an unavoidable factor. Local area networks (LAN) are an important tool for its implementation [1]. They are at the heart of supporting business life and day-to-day business operations, ensuring data reliability and the necessary information awareness [2].

Computer networks create a stable infrastructure that ensures communication, data sharing [6], and increased productivity in every respect. They have many advantages, one of which is the fast transfer of files or data sets [7], while maintaining data integrity and ensuring reliability in terms of cyber security measures. By implementing stable access control, we work towards data security, but fast transfer, in turn, sets requirements for high data transfer speeds and reflects on the rationalization of data sharing processes.

[1]Through network monitoring tools, administrators can closely monitor network traffic and its accompanying parameters and characteristics. Naturally, networks are growing in size, diversity, and complexity, so network administrators face many challenges in managing them skillfully. One of the most important issues related to the administration of a network is scalability, i.e. the network's ability to handle increasing network traffic [4], users, devices, and applications without compromising performance, security, or reliability. Scalability depends on various factors. Of course, bandwidth, routing protocols, node load, and more matter. But, optimizing network resources [10] and avoiding congestion, bottlenecks and failures is always related to the type of network topology. Against this background, the present work proposes a complex solution for analyzing network bottlenecks with a view to optimizing its scalability by enabling dynamic and centralized traffic management through real-time adaptability. The idea is to provide flexibility, efficiency of hierarchy of layers or levels with different functions and responsibilities, and standardization in network design to ensure interoperability.

The solution strategy includes observation, computational process, and optimization decision process. Monitoring includes capturing traffic data. The computing process provides information about performance and potential problem areas. The decision process plays the role of an optimizer of network operations and resources. The solution itself belongs to the realm of vertical scalability, aiming to improve the system's performance [9] and capacity to handle larger workloads or more complex tasks without changing the underlying architecture or adding additional devices.

II. TASK STATEMENT

The essence of the proposed solution is the idea of minimizing the average time delay of packets in a computer network by means of the application of algorithm for flexible optimization of routes [5] by minimizing time delays along them. In fact, each node of a computer network exchanges information with its neighbors and maintains its own routing table [8]. This table contains one field for each node in the network, which consists of at least two parts: a preferred baseline for reaching a neighboring node and some estimate of a system-observed parameter. Such a parameter for the purposes of our algorithm is the network packet delay time, estimated in milliseconds (ms).

Consider an example topology of a LAN given in Fig.1. The route traffics (RT[.]) on selected communication routes between nodes are shown in Table 1.

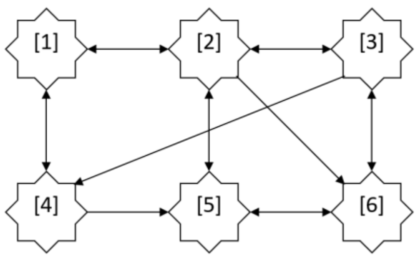

Fig. 1. Topology of LAN

Received: 19.09.2024
Published: 30.09.2024
https://doi.org/10.47978/TUS.2024.74.03.007

Nikolay Dimov is with Technical University of Sofia, Faculty and College – Sliven (ndimov@melinvest.bg)

Dimitar Nyagolov is with Technical University of Sofia, Faculty and College – Sliven (d_nyagolov@abv.bg)

Rossen Krystev is with Technical University of Sofia, Faculty of Computer Systems and Technologies (rosen20103@gmail.com)

TABLE 1
Route Traffic RT[.] on selected route between nodes

	[1]	[2]	[3]	[4]	[5]	[6]
[1]		10 RT[1-2]	8 RT [1-2-3]	8 RT [1-4]	7 RT [1-2-5]	9 RT[1-2-3-6]
[2]	8 RT [2-1]		4 RT [2-3]	2 RT [2-1-4]	5 RT [2-5]	4 RT [2-6]
[3]	4 RT [3-4-1]	9 RT [3-2]		3 RT [3-4]	11 RT [3-4-5]	7 RT [3-6]
[4]	4 RT [4-1]	6 RT [4-5-2]	2 RT [4-5-6-3]		12 RT [4-5]	5 RT [4-5-6]
[5]	31 RT [5-2-1]	3 RT [5-2]	1 RT [5-6-3]	36 RT[5-2-1-4]		3 RT [5-6]
[6]	6 RT[6-3-2-1]	3 RT [6-5-2]	26 RT [6-3]	2 RT [6-3-4]	4 RT [6-5]	

Table 1 shows, for example, the route traffic from node [1] to node [3] via node [2] is 8 packets, i.e. RT[1-2-3] = 8, whereas selected route between them is indicated as [1-2-3].

III. ALGORITHM OF THE COMPUTING PROCESS

Let the network bandwidth be $C = 10000\ bit/\sec$.

Let the average length in bits of the transmitted packets be L=100 bits. Then the frequency of packets served per second on every line will be defined as

$$\mu = C / L = 100\ \ packets/\sec .$$

Except the packets served in 1 sec. (μ) in each line we can calculate and the number of packets transmitted on every single line [i-j] in 1s (λ[i-j]). For example, for line [1-2], sum the packets of all routes using that line (see first row of Table 1) and get 10+8+7+9, which is λ[1-2] = 34 packets (see Table 2). Similarly, the number of transmitted packets for each of the other single lines is determined. So, we can calculate the average delay time for a line DL[.] in ms per packet for each of the lines using Little's formula from mass service theory for the average dwell time:

$$DL[i-j] = \frac{10^3}{\mu - \lambda[i-j]},\ \ ms\ .$$

TABLE 2
Lines and their delays of line DL[.] on LAN

Lines	$\lambda_{[i-j]}$, $(p.)$	C, (bit/s)	$\mu = C/L$, $(packet/s)$	$DL[i-j]$, (ms)
[1-2]	34	10 000	100	15,6
[1-4]	46	10 000	100	18,5
[2-1]	83	10 000	100	58,8
[2-3]	21	10 000	100	12,7
[2-5]	12	10 000	100	11,4
[2-6]	4	10 000	100	10,4
[3-2]	15	10 000	100	11,8
[3-4]	20	10 000	100	12,5
[3-6]	7	10 000	100	10,8
[4-1]	8	10 000	100	10,9
[4-5]	36	10 000	100	15,6
[5-2]	79	10 000	100	47,6
[5-6]	11	10 000	100	11,2
[6-3]	35	10 000	100	15,4

In this situation, it is easy to calculate the delay times DL[.] of the lines in our example network (see Table 2).

The goal of our work is to replace the current selected routes (Table 1) with better ones for which the time delays are optimally best.

First step: for the optimization needs in our algorithm, we will calculate the time delay matrix from each node to the other nodes. The current routing table (Table 1) can be used to calculate the delay times for each route that packets travel, which can be seen in Table 3.

TABLE 3
Delay times DR[.] for each route that packets travel

	[1]	[2]	[3]	[4]	[5]	[6]
[1]		DR[1-2] =15,6	DR[1-2-3] = 28,3	DR[1-4] =18,5	DR[1-2-5] =27	DR[1-2-3-6] =39,1
[2]	DR[2-1] =58,8		DR[2-3] =12,7	DR[2-1-4] =77,3	DR[2-5] =11,4	DR[2-6] =10,4
[3]	DR[3-4-1] =23,4	DR[3-2] =11,8		DR[3-4] =12,5	DR[3-4-5] =28,1	DR[3-6] =10,8
[4]	DR[4-1] =10,9	DR[4-5-2] =63,2	DR[4-5-6-3] =42,2		DR[4-5] =15,6	DR[4-5-6] =26,8
[5]	DR[5-2-1] =106,4	DR[5-2] =47,6	DR[5-6-3] =26,6	DR[5-2-1-4] =124,9		DR[5-6] =11,2
[6]	DR[6-3-2-1] =86	DR[6-5-2] =58,4	DR[6-3] =15,4	DR[6-3-4] =27,9	DR[6-5] =10,8	

Every delay time for a route is calculated following the sequence of lines encoded by the nodes in this route. For example, the delay time for the route DR[1-2-3] from node [1] to the node [3] via node [2] (see row [1], column [3] of Table 3) will be calculated by summing the time delays along the lines [1-2] and [2-3] (see Table 2), that is

DR[1-2-3] = DL[1-2] + DL[2-3] = 15,6+12,7 = 28,3

whereas the delay time for the route DR[5-2-1-4] from node [5] to the node [4] via nodes [2] and [1] (see row [5], column [4] of Table 3), is calculated by summing the time delays along the lines [5-2], [2-1] and [1-4] (see Table 2), that is

DR[5-2-1-4] = DL[5-2] + DL[2-1] + DL[1-4]
= 47,6+58,8+18,5 = 124,9

Proceeding in this way we can calculate the time delays for each route that packets travel, i.e. to obtain the matrix in Table 3. Now we are ready for the second stage of the algorithm.

IV. IMPROVEMENT OF THE ROUTING

Second stage: using the matrix in Table 3 as a base, we start a procedure to obtain a new plan for selection of communication routes between nodes with minimized time delays from each node of the network to the other nodes.

The procedure consists of a finite number of steps, following the lines of Table 2. For first step, the time delays of leading lines with a start point, associated with each particular node of the network, will be used (see Table 2), i.e. DL[1-2], DL[2-1], DL[3-2], DL[4-1], DL[5-2],DL[6-3]. Each of these line time delays will be used to replace an existing route to the corresponding node with another one with a smaller time delay. For example, DL[1-2] will be collected with every delay for the routes DR[2-?], situated in the second row of Table 3, because these delays belong to routes starting with "2", which is the second number in the signature of the delay of the line, i.e. DL[1-2]. The number obtained after this addition will be compared to the time delay of the current route, and if it is less, then the current route will be replaced, along with its corresponding time delay.

Let us demonstrate how the first row of Table 3 will be changed by DL[1-2] on the first step.

DL[1-2]+DR[2-1]=15,6+58,8=74,4.

Here, the obtained route is [1-2-1], which is pointless, because the route "starts from node [1] and finishes with the node [1] and definitely there is no traffic at all. This is related to the first (which is empty) cell in the first row of Table 3.

Obviously, DL[1-2]=DR[1-2] and nothing will be changed in the second cell of the first row of Table 3.

DL[1-2]+DR[2-3]= 15,6+12,7=28,3=28,3=DR[1-2-3].

The resulting route is [1-2-3] and its time delay is equal to the time delay of the currently active route from Table 3, i.e. DR[1-2-3]. That is why, there is no need to change it.

DL[1-2]+DR[2-1-4]= 15,6+77,3=92,9>18,5=DR[1-4].

The resulting route is [1-2-1-4] and its time delay is greater than the time delay of the currently active route from Table 3, i.e. DR[1-4]. That is why, there is no need to change this route.

DL[1-2]+DR[2-5]= 15,6+11,4=27=DR[1-2-5].

The resulting route is [1-2-5] and its time delay is equal to the time delay of the currently active route from Table 3, i.e. DR[1-2-5]. That is why, there is no need to change this route.

DL[1-2]+DR[2-6]= 15,6+10,4=26<39,1=DR[1-2-3-6].

Here, the resulting route is [1-2-6] and its time delay is less than the time delay of the currently active route from Table 3, i.e. DR[1-2-3-6]. That is why, we have to change the route DR[1-2-3-6] with the route DR[1-2-6].

Therefore, the first row of Table 3 will be changed by DL[1-2] only at one place, namely at DR[1-2-3-6]. At this place a new one route is chosen, i.e. DR[1-2-6], where the route traffic from node [1] to node [6] via node [2] is provided more efficiently in terms of time delay.

Similarly, DL[2-1] will try to change the contents of the second row of Table 3, adding itself to the delay times of all routes departing from node [1]. So, we have:

DL[2-1]=DR[2-1].

DL[2-1]+DR[1-2-3]=58,8+28,3=87,1>12,7=DR[2-3].

DL[2-1]+DR[1-4]= 58,8+18,5=77,3=77,3=DR[2-1-4].

DL[2-1]+DR[1-2-5]=58,8+27=85,8>11,4= DR[2-1-5].

DL[2-1]+DR[1-2-6]=58,8+26=84,8>10,4= DR[2-6].

Obviously, the second row of Table 3 will remain unchanged by DL[2-1].

Further, DL[3-2] will try to change the contents of the third row of Table 3, adding itself to the delay times of all routes departing from node [2]. So, we have:

DL[3-2]+DR[2-1]=11,8+58,8=70,6>23,4=DR[3-4-1].

DL[3-2]=DR[3-2].

DL[3-2]+DR[2-1-4]=11,8+77,3=89,1>12,5=DR[3-4].

DL[3-2]+DR[2-5]=11,8+11,4=23,2<28,1= DR[3-4-5].

DL[3-2]+DR[1-2-6]=58,8+26=84,8>10,4= DR[2-6].

Therefore, the third row of Table 3 will be changed by DL[3-2] only at one place, namely at DR[3-4-5]. At this place a new one route is chosen, i.e. DR[3-2-5], where the route traffic from node [3] to node [5] via node [2] is provided more efficiently in terms of time delay.

Further, DL[4-1] will try to change the contents of the fourth row of Table 3, adding itself to the delay times of all routes departing from node [1]. So, we have:

DL[4-1]=DR[4-1]

DL[4-1]+DR[1-2]=10,9+15,6=26,5<63,2=DR[4-5-2].

DL[4-1]+DR[1-2-3]= 10,9+28,3=39,2<42,2=DR[4-5-6-3].

DL[4-1]+DR[1-2-5]= 10,9+27=37,9>15,6= DR[4-5].

DL[4-1]+DR[1-2-3-6]= 10,9+39,1=50>26,8= DR[4-5-6].

Therefore, the fourth row of Table 3 will be changed by DL[4-1] at two places: DR[4-5-2] and at DR[4-5-6-3]. At DR[4-5-2] a new route is chosen, i.e. DR[4-1-2], where the route traffic from node [4] to node [2] via node [1] is provided more efficiently in terms of time delay. Also, at DR[4-5-6-3] a new route is chosen, i.e. DR[4-1-2-3], where the route traffic from node [4] to node [3] via nodes [1] and [2] is provided more efficiently in terms of time delay.

Further, DL[5-2] will try to change the contents of the fifth row of Table 3, adding itself to the delay times of all routes departing from node [2]. So, we have:

DL[5-2]+DR[2-1]=47,6+58,8=106,4=DR[5-2-1].

DL[5-2]=DR[5-2].

DL[5-2]+DR[2-3]= 47,6+12,7=60,3>26,6=DR[5-6-3].

DL[5-2]+DR[2-1-4]= 47,6+77,3=124,9= DR[5-2-1-4].

DL[5-2]+DR[2-6]= 47,6+10,4=58>11,2= DR[5-6].

Obviously, the fifth row of Table 3 will remain unchanged by DL[5-2].

Finally, DL[6-3] will try to change the contents of the sixth row of Table 3, adding itself to the delay times of all routes departing from node [3]. So, we have:

DL[6-3]+DR[3-4-1]=15,4+23,4=38,8<86=DR[6-3-2-1].

DL[6-3]+DR[3-2]= 15,4+11,8=27,2<58,4=DR[6-5-2].

DL[6-3]=DR[6-3].

DL[6-3]+DR[3-4]= 15,4+12,5=27,9= DR[6-3-4].

DL[6-3]+DR[3-4-5]= 15,4+28,1=43,5>10,8= DR[6-5].

Therefore, the sixth row of Table 3 will be changed by DL[6-3] at two places: DR[6-3-2-1] and at DR[6-5-2]. At DR[6-3-2-1] a new route is chosen, i.e. DR[6-3-4-1], where the route traffic from node [6] to node [1] via nodes [3] and [4] is provided more efficiently in terms of time delay. Also, at DR[6-5-2] a new route is chosen, i.e. DR[6-3-2], where the route traffic from node [6] to node [2] via node [3] is provided more efficiently in terms of time delay.

Thus, Table 3 will be improved to Table 4 with some routes with smaller time delays.

TABLE 4
Improved delay times DR[.] at first step of Stage 2

	[1]	[2]	[3]	[4]	[5]	[6]
[1]		DR[1-2] =15,6	DR[1-2-3] = 28,3	DR[1-4] =18,5	DR[1-2-5] =27	DR[1-2-6] =26
[2]	DR[2-1] =58,8		DR[2-3] =12,7	DR[2-1-4] =77,3	DR[2-5] =11,4	DR[2-6] =10,4
[3]	DR[3-4-1] =23,4	DR[3-2] =11,8		DR[3-4] =12,5	DR[3-2-5] =23,2	DR[3-6] =10,8
[4]	DR[4-1] =10,9	DR[4-1-2] =26,5	DR[4-1-2-3] =39,2		DR[4-5] =15,6	DR[4-5-6] =26,8
[5]	DR[5-2-1] =106,4	DR[5-2] =47,6	DR[5-6-3] =26,6	DR[5-2-1-4] =124,9		DR[5-6] =11,2
[6]	DR[6-3-4-1] =38,8	DR[6-3-2] =27,2	DR[6-3] =15,4	DR[6-3-4] =27,9	DR[6-5] =10,8	

Proceeding in a similar way, in the second step we will use the following time delays of lines, which follow those used in the first step (see Table 2), i.e. DL[1-4], DL[2-3], DL[3-4], DL[4-5], DL[5-6]. Further, in the next step we will use the following time delays of lines, which follow those, chosen at second step (see Table 2), i.e. DL[2-5], DL[3-6]. In the last step we will use the rest time delay of line, i.e. DL[2-5].

At each of these steps, the table of time delays along the various routes (of the form of Table 4) will be improved with increasingly minimal time delays along these routes. So, finally we obtain the optimized route table with minimized time delays for routes, see Table 5.

TABLE 5
Improved delay times DR[.] at first step of Stage 2

	[1]	[2]	[3]	[4]	[5]	[6]
[1]		DR[1-2] =15,6	DR[1-2-3] = 28,3	DR[1-4] =18,5	DR[1-2-5] =27	DR[1-2-6] =26
[2]	DR[2-3-4-1] =36,1		DR[2-3] =12,7	DR[2-3-4] =25,2	DR[2-5] =11,4	DR[2-6] =10,4
[3]	DR[3-4-1] =23,4	DR[3-2] =11,8		DR[3-4] =12,5	DR[3-6-5] =21,6	DR[3-6] =10,8
[4]	DR[4-1] =10,9	DR[4-1-2] =26,5	DR[4-1-2-3] =39,2		DR[4-5] =15,6	DR[4-5-6] =26,8
[5]	DR[5-6-3-2-1] =98,2	DR[5-2] =47,6	DR[5-6-3] =26,6	DR[5-6-3-4] =40,1		DR[5-6] =11,2
[6]	DR[6-3-4-1] =38,8	DR[6-3-2] =27,2	DR[6-3] =15,4	DR[6-3-4] =27,9	DR[6-5] =10,8	

V. CONCLUSIONS

The conducted research shows that time delays in a computer network can be minimized by applying appropriate algorithms.

The applied methodology and researches show that in the process of operation with the means of automation, the optimal path for data transfer can be monitored and selected, thereby increasing the efficiency of the computer networks.

Let us note that our proposed algorithm belongs to the group of centralized adaptive algorithms. In centralized adaptive algorithms, a single routing control center is created in the network. It calculates the routing tables of all nodes and sends them to them. In order to adapt the routing tables to the current topology and current traffic, all nodes must send information to the routing center. Based on the received information, the routing center calculates the weights of the edges and then calculates the optimal route between any two nodes. It is good, of course, to maintain alternate paths between nodes. In addition, it is good to consider that the information from nodes closer to the route center will arrive faster than from those further away. For this reason, the update period of the routing tables must be at least twice the time for a packet to travel from the routing center to the node most distant from it.

A recalculated routing table received at one node should not be used immediately, as routing tables arrive at different times at different nodes.

If for some reason the routing center goes down, the network is left unmanaged. For this purpose, the route center could be duplicated, but then the service traffic would increase too much.

ACKNOWLEDGMENT

The authors would like to thank Donev, V. I., PhD, for valuable guidance and recommendations in the smoothing and formalization of the algorithm presented in this work.

REFERENCES

[1] Bachvarov A. at al., Proektirane na AIS, publ. house “Nauka i izkustvo”, Sofia, 1989 (in Bulgarian).
[2] Iliev G., Atamyan D., „Mrezhi za danni i internet comunikacii“, publ. house “Novi znania”, Sofia, 2009 (in Bulgarian)
[3] Pulkov Vl., Koleva P., „Osnovi na predavane na informaciata“, publ. house “Novi znania”, Sofia, 2009 (in Bulgarian)
[4] Almeroth, K., The Evolution of Multicast: From the Mbone to Interdomain Multicast to Internet2 Deployment, IEEE Network, 2000
[5] Ash, G., Dynamic Routing in Telecommunications Networks, McGraw Hill, NY, 1998
[6] Bertekas D., Gallager R., Data Network, Englewood Cliffs, 1992
[7] Bertsekas, D., R. Gallager. Data Networks (2nd ed), Prentice Hall, Englewood Cliffs, NJ, 1992
[8] Black, U., IP Routing protocols: RIP, OSPF, BGP, PNNI & Cisco Routing Protocols. Prentice Hall, 2000
[9] https://obkio.com/blog/lan-monitoring-how-to-improve-lan-network-performance/
[10] https://daily.dev/blog/10-network-optimization-tips-to-boost-bandwidth