DOI: https://doi.org/10.5281/zenodo.11189619

ANALYSIS OF INDICATORS OF PACKET SERVICES OF THE NETWORK

Olimov Olimboy Otabekovich
Urgench branch of the Tashkent University of Information
Technologies named after Muhammad Al-Khwarizmi
E-mail: olimovolimboy274@gmail.com

Abstract: *Accurate and in-depth analysis of network performance is a complex process. Network functionality can only cover a subset of the ever-changing workloads that processes on the network experience, which can lead to unexplored node states and sub-optimal performance in live traffic. is considered Queuing theory and network calculus techniques can provide tight bounds on performance metrics, but typically require approximations of the state of network components and the traffic arrival order with short and well-executed mathematical functions. As such, they are not immediately applicable to emerging workloads and the new algorithms and protocols developed to handle them. We explore a different approach: using formal methods to analyze network performance. We show that application synthesis techniques can be used to accurately model network components and their queues and automatically generate concise interpretable workloads in response to performance metrics queries. Our approach offers a new point in the field of available tools for network performance analysis: it is more complete than simulation and emulation and can be easily applied to algorithms and protocols expressed in first-order logic. We demonstrate the effectiveness of our approach by analyzing packet scheduling algorithms and a small leaf-spine network and creating compact workloads that may suffer from throughput, fairness, starvation, and latency issues.*

Key Words: *OptQuest, Delay, Queue Sampler, Exit.*

1.Introduction: When used to solve quantitative analysis problems (for example, system efficiency indicators) in optimization experiments, finding alternative values of factor solutions is called the reverse process of simulation modeling. The inverse process is the correct assignment to solve the inverse problem several times when answering the problems of showing the solutions that provide the maximum efficiency of the system among the available solutions. In cases where the number of solution options is not large, the inverse process puts all the solutions using a simple "try it out" method and compares them to find the optimal solution[1].

If it is not possible to try all solutions, then methods based on selection using heuristics are used[2]. In this case, the optimal or near-optimal solutions are found after several successive steps (finding the solution of the correct task and creating a vector of output results for each input parameter). A well-chosen heuristic brings the experiment closer to the optimal solution at every step[3].

The user can use any external optimizer or OptQuest optimization software included in the package as a block for recording output results and for further optimization (Figure 1).

The OptQuest optimizer uses metaheuristic scatter search and tabu search. Currently, this optimizer is considered one of the best among professional optimization packages for solving complex optimization problems [4-5].

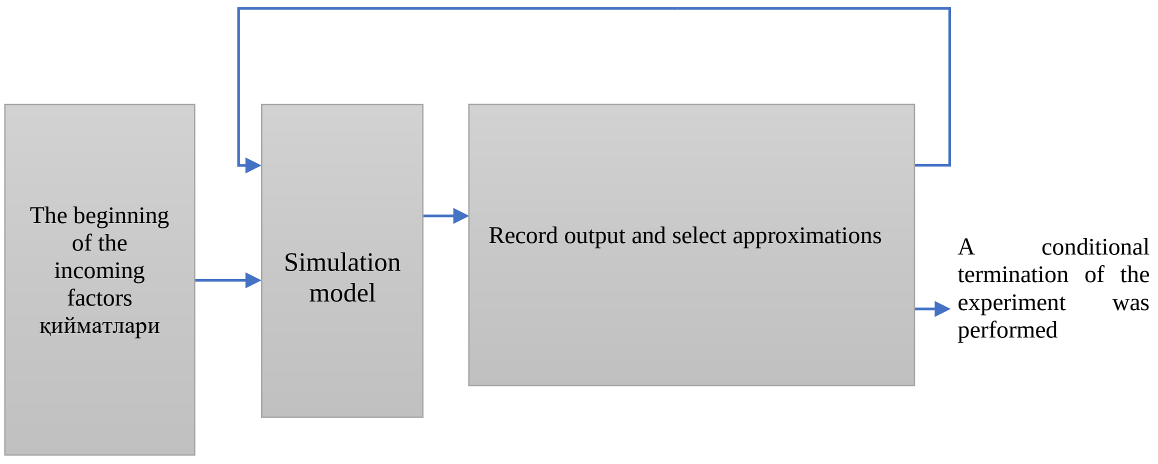

The beginning of the incoming factors қийматлари
Simulation model
Record output and select approximations
A conditional termination of the experiment was performed

Figure 1. Functional diagram of the OptQuest optimization software

2.OptQuest runs directly in the model development environment:

To configure the optimization, you need to do the following in AnyLogic[6].

1) Creating optimization experience for the developed model;

2) Enter optimization parameters and their change range;

3) Include a condition to stop modeling after each run. This can be done by stopping in time or by conditioning on the values of model variables;

4) Entering the target, i.e., the system reaction function to be studied;

5) At the end of each run, selection of restrictions, which will determine the compatibility of the value of the waiting vector[7];

6) Enter the condition of stopping the experiment.

Once the model is loaded, the optimization experiment selects alternative input parameters that are minimal or maximal based on the given function[3].Minimizing the delay (backlog) of orders (packages) in the public service system is of great importance.Mathematical representation of this problem:

$$min \sum_{i=1}^{n} Vt_{zi} \qquad (1)$$

$$\sum_{i=1}^{n} V_i \leq V_{\sum} \qquad (2)$$

tz-is the average delay time of the order on the i-th device (server)[8-9];

Vi- i- server speed;

V∑-is the overall speed of the servers.

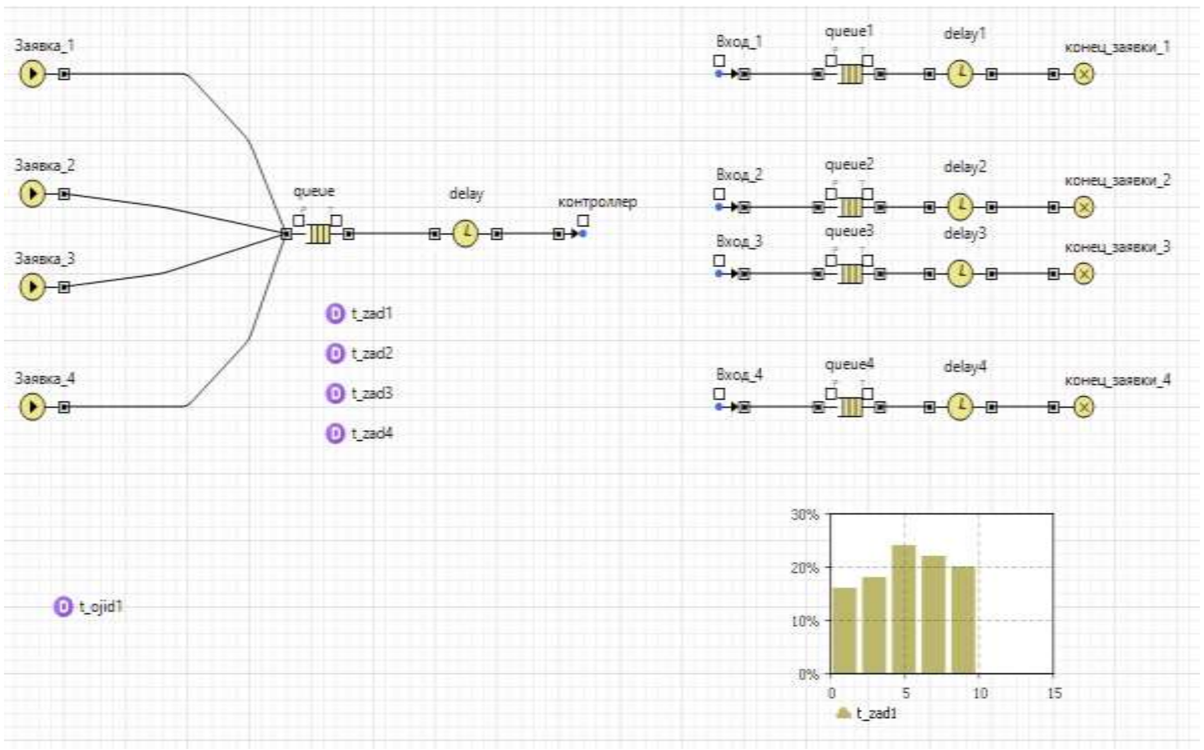
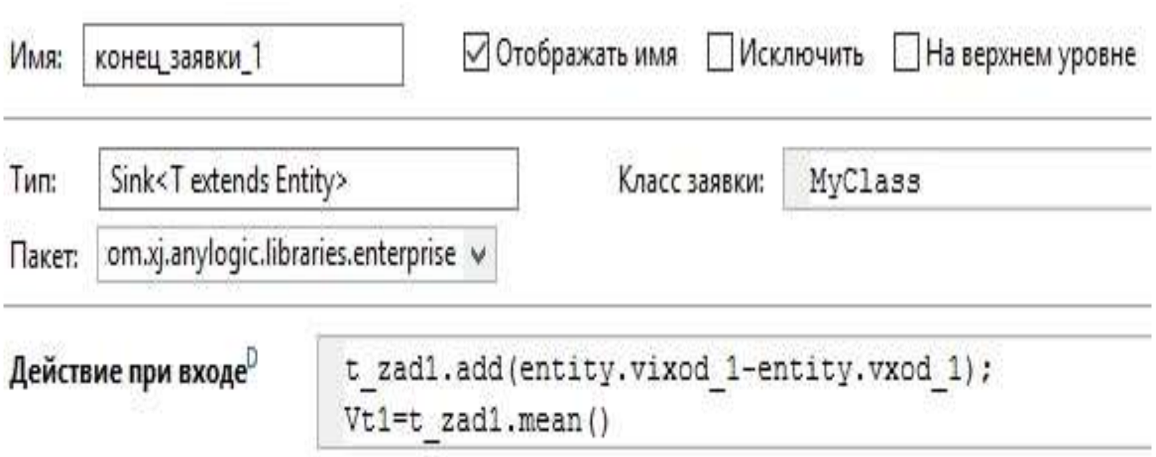


Figure 2. OXKT model.

Let's create a histogram to calculate the order delay:



Имя: конец_заявки_1 ☑ Отображать имя ☐ Исключить ☐ На верхнем уровне

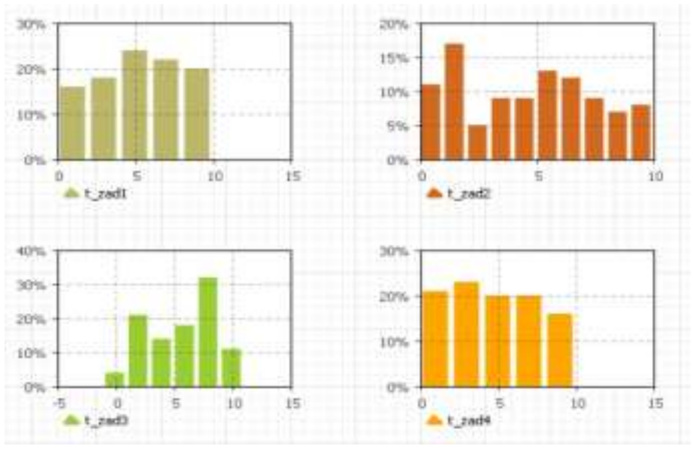
Тип: Sink<T extends Entity> Класс заявки: MyClass

Пакет: om.xj.anylogic.libraries.enterprise

Действие при входе

```
t_zad1.add(entity.vixod_1-entity.vxod_1);
Vt1=t_zad1.mean()
```

Figure 3. Create a histogram of latency.



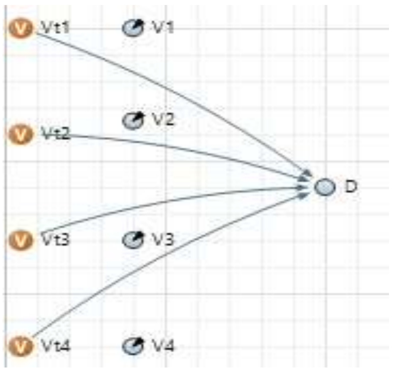
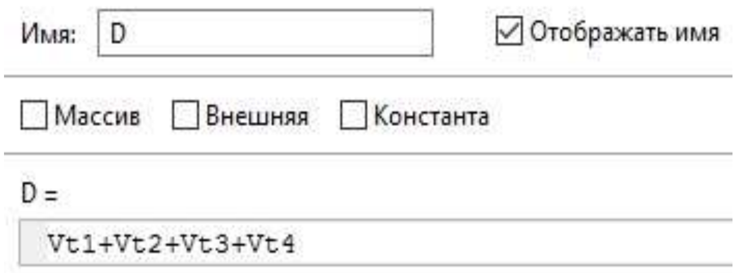


Figure 4. Latency histograms. Figure 5. Optimization model.



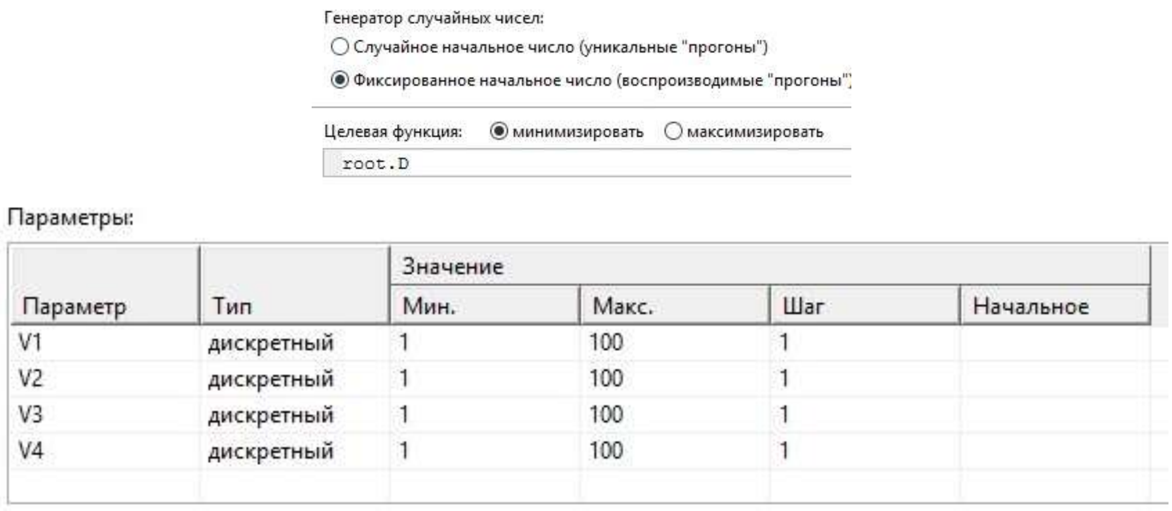
Имя: D ☑ Отображать имя

☐ Массив ☐ Внешняя ☐ Константа

D =

```
Vt1+Vt2+Vt3+Vt4
```

Figure 6. Enter the objective function.



Генератор случайных чисел:
○ Случайное начальное число (уникальные "прогоны")
[✓ Фиксированное начальное число (воспроизводимые "прогоны")]

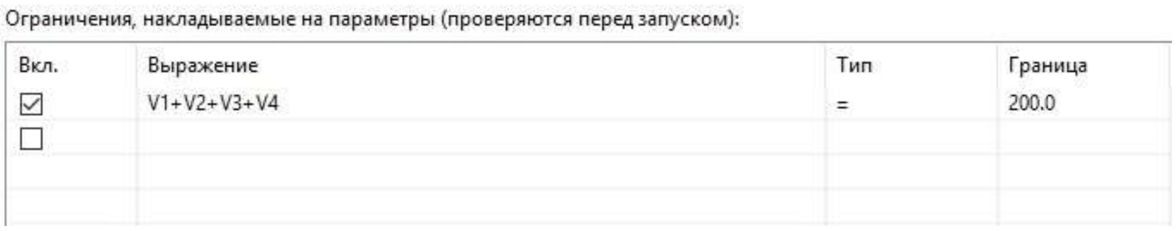
Целевая функция: [✓ минимизировать] ○ максимизировать

root.D

Параметры:

Параметр	Тип	Значение			
		Мин.	Макс.	Шаг	Начальное
V1	дискретный	1	100	1	
V2	дискретный	1	100	1	
V3	дискретный	1	100	1	
V4	дискретный	1	100	1	

Figure 7. Enter optimization parameters.



Ограничения, накладываемые на параметры (проверяются перед запуском):

Вкл.	Выражение	Тип	Граница
✓	V1+V2+V3+V4	=	200.0
☐			

Figure 8. Introduction of optimization steps.

Determination of efficiency optimization

Initial parameters: $\lambda_1 = 10$, $\lambda_2 = 40$, $\lambda_3 = 20$, $\lambda_4 = 50$, $\lambda_k = 200$ i $V_{vch1} = V_{vch2} = V_{vch3} = V_{vch4} = 50$.

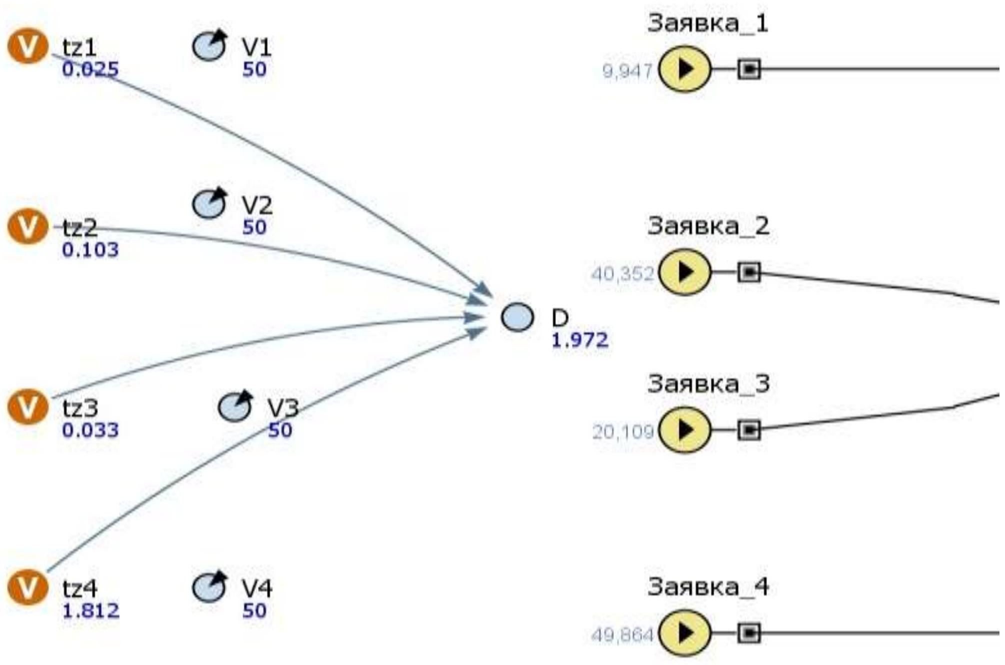
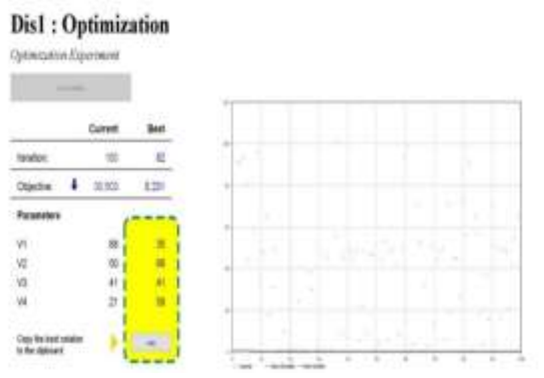


Figure 9. Results before optimization.

The results obtained after the optimization experiment are presented in Fig. 9.

Figure 10. Optimization results.



Dis1 : Optimization

Optimization Experiment

	Current	Best
Iteration:	100	82
Objective:	[illegible]	[illegible]

Parameters

V1	88	[illegible]
V2	60	66
V3	41	41
V4	21	58

Copy the best solution to the clipboard

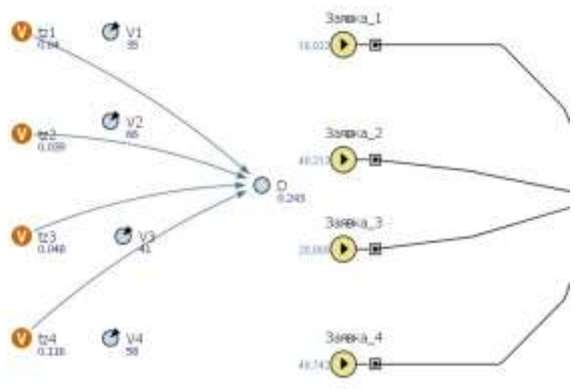


Figure 11. Optimized parameter values.

3. Workload metrics. The search algorithm explores the space of all workloads to find one that satisfies the query. When synthesizing candidate workloads, it decides how many constraints to include in the workload, and what metrics and queues to include in each constraint [10].

While we leave the metrics that can be used in queries unconstrained, deciding the set of metrics that are used in synthesizing workloads requires careful consideration. We want the set to be small to keep the search space tractable but expressive to enable specifying common workloads in a concise and intuitive manner[11-12]. We define our workloads over two metrics: cumulative enqueues (cenq(q, t)), the total number of packets that enter q by the end of time step t, and arrival inter-packet gap (aipg(q, t)), the inter-packet gap between the last two packets that enter q by time t.

While small, this is a quite expressive set. cenq constrains the total number of packets entering the queue, independent of the exact time they arrive. So, it can abstract away the timing details of traces when they are not important for the query. aipg, on the other hand, constrains the gap between packets and their arrival pace. So, it can capture low-level timing details if necessary in answering the query. Together, they create a good balance in capturing the commonalities of traces that satisfy a query, abstracting away unnecessary details and including necessary ones[13-14].

Our experience in the case studies has shown that this set is capable of expressing a variety of workloads. But, we view this as a suitable starting point and not necessarily the final answer. We hope that as using formal methods, and specifically workload synthesis, for performance analysis evolves, the set of metrics will mature as well. In fact, our search algorithm is parametrized over the set of metrics and, if needed, any metric that can be encoded in SMT can be easily added to our search algorithm.

4.Synthesizing Answers

The search engine uses a guided randomized search over the space of workloads to find one that satisfies the query. The search algorithm (Algorithm 1) is based on the Metropolis Hastings Markov Chain Monte Carlo (MCMC) sampler, which combines random walks with hill climbing and has been successfully used for synthesizing optimized programs and loop invariants [15]. Starting from an initial workload wl = true (line 3), which imposes no constraints on the input queues, the search algorithm asks the verification engine to verify the workload, i.e., check if all the traces in the workload satisfy the query (line 6). If yes, the search engine returns the workload as the answer to the query (line 7). If not, using the feedback from the verification engine, the search algorithm moves on to synthesize and try another candida teworkload until a suitable workload is found.

5.CONCLUSION

Over the past decade, a large body of academic and industry work has demonstrated the feasibility and benefits of using formal methods to reason about the functional correctness of networks. Inspired by their success, we set out to bring the same benefits to analyzing network performance.

Along the way, we have developed efficient encodings of packet-level interactions that affect network performance. We have also found that when it comes to performance analysis, returning isolated packet traces that violate performance properties is not always useful. Instead, we argue that a more useful output is a workload that can concisely describe the commonality of a set of traces that can experience performance problems. We have shown how to apply existing synthesis techniques to generating such workloads and demonstrated the tractability of our approach using case studies.

This is only the start; as with other applications of formal methods to systems and networking, much work needs to be done to make such formal performance analysis approaches suitable for analyzing real-world networks, some of which we have outlined in this paper as future research directions.

6.REFERENCES

[1] S. Avallone, S. Guadagno, D. Emma, A. Pescape, and G. Ventre, "D-ITG distributed Internet traffic generator", First International Conference on the Quantitative Evaluation of Systems, 2004.

[2] ZTI Telecom, "IP Traffic - test & measure," http://www.zti-telecom.com.

[3] S. Avallone, "Mtools 1.1," http://www.grid.unina.it/grid/mtools/, 2002.

[4] J. Laine, S. Saaristo, and R. Prior, "Rude & crude," http://rude.sourceforge.net/.

[5] Olimov O. O., Saparbayev R. K. Network traffic queue analysis //Educational Research in Universal Sciences. – 2024. – T. 3. – №. 2. – p. 311-318.

[6] https://cyberleninka.ru/article/n/analog-to-digital-conversion-process-by-matlab-simulink

[7] Naval Research Laboratory, "Multi-Generator (MGEN)," http://cs.itd.nrl.navy.mil/work/mgen/index.php.

[8] The University of Michigan, "gen_send, gen_recv: a simple uDP traffic generator application," http://www.citi.umich.edu/projects/qbone/generator.html.

[9] T. Lattner, D. Cook, and K. Gibbs, "Jperf,"http://dast.nlanr.net/projects/jperf/.

[10] A. K. Agarwal and W. Wang, "Measuring performance impact of security protocols in wireless local area networks," 2nd International Conference on Broadband Networks, 2005.

[11] T.-Y. Wu, H.-C. Chao, T.-G. Tsuei, and Y.-F. Li, "A measurement study of network efficiency for TWAREN IPv6 backbone," International Journal of Network Management, vol. 15, pp. 411-419, 2005.

[12] K. A. Gotsis, S. K. Goudos, and J. N. Sahalos, "A test lab for the performance analysis of TCP over Ethernet LAN on windows operating system," IEEE Transactions on Education, vol. 48, pp. 318-328, 2005.

[13] A. K. Agarwal, J. S. Gill, and W. Wenye, "An experimental
study on wireless security protocols over mobile IP networks," IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall.

[14] Olimboy O. Multiservice network services //Web of Humanities: Journal of Social Science and Humanitarian Research. – 2024. – T. 2. – №. 2. – p. 101-104.

[15] A. Botta, A. Dainotti, and A. Pescape, "Multiprotocol and multi-platform traffic generation and measurement," INFOCOM 2007 DEMO Session., Anchorage, Alaska, 2007.