



## Analysis and Design of a Hybrid Load Management method for the IoT Networks

Vitalii Savchenko<sup>1</sup>, Volodymir Druzhynin<sup>2</sup>, Mykola Tverdohlib<sup>3</sup>, Yevhen Ivanichenko<sup>4</sup>,  
Nadiia Dovzhenko<sup>5</sup>, Zoreslava Brzhevskia<sup>6</sup>, Valentina Chorna<sup>7</sup>

<sup>1</sup>Doctor of Technical Sciences, Professor, Director of the Information Security Institute,  
State University of Telecommunications, Ukraine, savitan@ukr.net

<sup>2</sup>Doctor of technical sciences, Professor, Professor of department of radio engineering and radioelectronic systems,  
Taras Shevchenko National University of Kyiv, Ukraine, druzhynin@ukr.net

<sup>3</sup>PhD, Associate Professor of the Department of Telecommunications and Networks,  
State University of Telecommunications, Ukraine, tverdohlib@ukr.net

<sup>4</sup>PhD in technical sciences, Senior Lecturer, Associate Professor of the Department of Energy Efficient Technologies, State  
University of Telecommunications, Ukraine, chhe@ukr.net

<sup>5</sup>PhD in technical sciences, Associate Professor, Department of Information and Cybersecurity,  
State University of Telecommunications, Ukraine, nadezhdadovzhenko@gmail.com

<sup>6</sup>Postgraduate student, Teachers Assistant at the Department of Information and Cybersecurity,  
State University of Telecommunications, Ukraine, zoreska.puzniak@gmail.com

<sup>7</sup>Senior Lecturer, Department of Information Systems and Technologies,  
State University of Telecommunications, Ukraine, chornav22@gmail.com

### ABSTRACT

The research paper consists of the simulation results of the hybrid algorithm for the IoT (The Internet of Things) networks based on the base load prediction. IoT is one of the most relevant and promising technology today. These networks consist not only of sensors or computer nodes but also they can support numerous applications. After deploying such a network, its elements are able to self-organize into an autonomous network that requires minimal interference. The previous research has shown that relaying a large number of requests across network elements leads to a decrease in functionality and an increase in bounce rates. However, the main task of research is to collect, process, store and transfer information from IoT devices to data centers or servers.

**Key words:** Internet of Things, flow, sensors, hybrid system, network, configuration, data center, information, protocol, server.

### 1. INTRODUCTION

#### 1.1 Literature analysis and problem statement

[1] The article provides an overview of existing IoT standards, protocols, and platforms. It is also mentioned that this technology is a great solution for many different types of applications, including healthcare, home and industrial automation because of its mobility, scalability and good management.

The aim goal of research [2] is cloud computing with IoTs, termed also as Cloud of Things (CoT). It provides ease of management for the growing media content and other data. Besides this, features like: ubiquitous access, service creation, service discovery, and resource provisioning play a significant role, which comes with CoT. Emergency, healthcare, and latency sensitive services require real-time response. The authors proposed a model which covers the issues of resource prediction, customer type based resource estimation and reservation, advance reservation, and pricing for new and existing IoT customers, on the basis of their characteristics.

In [3] the graph-oriented mechanism for achieving the smart transportation system in the city were presented. The authors proposed to deploy road sensors to get the overall traffic information. These IoT based networks generate enormous volume of data, termed as Big Data, depicting the traffic information of the city. The results show that the proposed system is more scalable and efficient.

The paper [4] investigates the process of optimization of network flow distribution based on the traffic engineering method. This method allows the use of separate measurement paths, reducing the likelihood of the overflow of measuring structures. The features of traffic engineering were considered and investigated by using a math model, which can be used.

Cloud Computing mentioned in this work [5]. The study focused on the research requirements and the basics for

majorly two categories – the learners and the researchers exploring the needs of users to cater by cloud and also the current research trends in cloud computing.

The authors in [6] assess the various machine learning methods that deal with the challenges presented by IoT data by considering smart cities as the main use case. The presentation of a taxonomy of machine learning algorithms explaining how different techniques are applied to the data in order to extract higher level information are presented.

This paper [7] offers comprehensive insights towards the usage of the multicast routing scheme with respect to offering security solution in the future wireless environment. The study finding of this investigation shows that there has been extensive usage of the multicast scheme, but the applicability of such scheme for a futuristic wireless communication system is highly restrictive as well as there was lack of consideration of dynamicity characteristics of adversaries. The paper also contributes to highlighting the open research issues followed by significant points that are necessary to be considered while developing a secure multicast routing scheme in the future.

In this report [8] a novel framework is developed based on cloud and internet of things that derives a new business model by integrating heterogeneous applications and devices. The integrated applications and devices are monitored in real time to provide smart services to end users in case of malfunction or repair of appliances to provide a suitable solution.

The comprehensive and flexible architecture based on distributed computing platform for real-time traffic control described in the article [9]. The architecture is based on systematic analysis of the requirements of the existing traffic control systems. Authors realized the architecture in a prototype platform that employs Kafka, a state-of-the-art Big Data tool for building data pipelines and stream processing. The approach on a case study of controlling the opening and closing of a freeway hard shoulder lane in microscopic traffic simulation was presented.

[10] The scientists propose a big data processing design for smart city, which will make good use of the collected data by adding a data modeling approach that treats both the transmission and the processing data, the others levels will be processed in other papers.

This paper [11] presents a framework for the virtualization of real world objects and the cognitive management of their virtual copies. It consists three levels of functionality and each level comprises cognitive entities that provide the means for self-management and learning, allowing for smart, flexible applications and objects. The paper also presents a corresponding prototype that has been developed for the validation of the proposed approach, in a real-life fire detection scenario in a Smart Home.

So, the importance of load management for IoT networks is important and relevant. Data centers and servers are central elements because of the main load transfer on them. So it is really important to analyze and design approach which will help to increase the flexibility of managing IoT network elements.

## 1.2 The aim of research

When using traditional IoT network monitoring methods, raw, redundant service requests are constantly generated. This reduces network performance and adversely affects channel performance. Therefore, it is proposed to apply a mechanism for dynamically changing the intensity of the control of the state of network elements, in which, depending on the difference between the values of the predicted and actual load conditions and resource management.

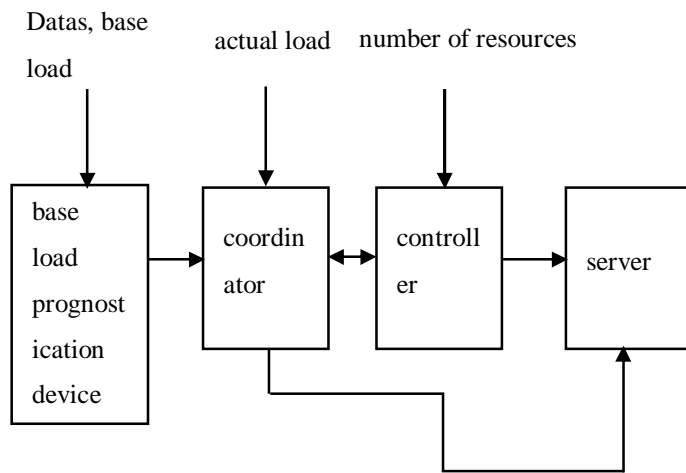
By using this approach, it is possible to increase the flexibility of managing network elements. As a consequence, the resources of the control devices will be used more efficiently. In addition, faster processing of service requests and keeping up-to-date information enables optimal load management throughout the network. This will significantly reduce electricity costs and increase the lifespan of IoT network elements.

## 2. THE MAIN SECTION

The idea of a hybrid load management system is that fixed and continuous load models are recorded based on previously obtained and analyzed data (base load). Thanks to this, algorithms for forecasting and managing network resources to serve it can be implemented. Therefore, in the event of a significant difference between the actual load and the intended load, all requests are processed in a hybrid way.

Figure 1 shows a developed scheme that considers the costs and risks associated with managing resources (depicted dependencies are schematic and used as an illustrative example) and is based on the classic architecture:

1. base load prognostication device - analyzes the previously obtained load data and determines the patterns that form the base load.
2. coordinator - sends load requests to remote servers, as well as communicates with the controller to provide input load information.
3. controller - estimates and allocates the appropriate amount of resources needed to handle the base load and excess requests.



**Figure 1:** Hybrid resource management system

Internet of Things (IoT) network needs traffic aggregation points. The main function of these points is to send specialized requests to the network equipment. These requests relate to a dedicated service chain that can be implemented by virtual network functions. It also should be noted that each function has its own quality of service (QoS) requirements.

For ease of implementation, it can be supposed that there is only one type of resource in the system. To obtain a mechanism for the dynamic allocation of resources, the problem must be formulated. Formally,  $d$  – the target response time of a network function and  $T$  – the average response time. It is necessary to allocate resources for a network function to satisfy:  $T \leq d$ . Expression (1) describes the principle of changing the frequency of monitoring of states:

$$W(t) = I_{base} - K \sum_{j=t-h}^{t-1} \frac{\max(0; \lambda_{obs}(j) - \lambda_{pred}(j))}{h} I_{base}, \tag{1}$$

where  $W$  – the interval during which the corresponding defined resources will be allocated,  $I_{base}$  – the base value of the interval calculated according to the following method of discretization of the load,  $K$  – the normalization constant, determined accordingly by the network operator,  $\lambda_{obs}(t)$  – the real intensity of the load flow over interval  $t$ ,  $\lambda_{pred}(t)$  – the predicted intensity of load flow during interval  $t$ ,  $h$  – number of elapsed intervals considered by the algorithm.

Next, a base interval needs to be defined. The purpose of this step is to present a daily pattern in the workload, sampling its queries at successive, non-consecutive time intervals with a single representative value at each interval.

An algorithm that helps to find a small number of time intervals is proposed. An important feature is that the deviation from the actual requirement would be minimized. Maintaining a small number of intervals is important because

as more intervals are inline mean more frequent changes in resource allocation. This will result in higher risks and costs.

It is necessary to define a formal sampling. Load sampling will consists time series  $X$  in the interval  $[v, T]$ , time series  $Y$  in the same interval. If  $[v, T]$  sampling of the load  $X$  can be divided into  $m$  consecutive intervals,  $\{[v, T_1], [T_1, T_2], \dots, [T_{m-1}, T]\}$ , so that  $X(j) = r_i$  for all  $j$  in the  $i$  – interval,  $[T_{i-1}, T_i]$ . Set  $v = 0$  and let  $T$  be the load period. Next we assume a period of 8 hours.

The idea of sampling has two sides. Firstly, we need to represent the load accurately. To achieve this, the representative values for  $r_i$  of each interval,  $[T_{i-1}, T_i]$  should be as close as possible to the actual values of the time series in the interval  $[T_{i-1}, T_i]$ . Secondly, IT resources are not allocated free of charge.

For this reason, too many intervals and, therefore, too many changes to the system should be avoided, as this is not practical and can lead to many problems (such as performance loss, server degradation, system instability, etc.). Thus, the sampling errors and the number of sampling intervals should be minimized. A solution for time series sampling (to reduce both) is proposed.

$$\sum_{i=1}^m \sum_{T=T_{i-1}}^{T_i} v(r_i - X(T)) + f(m) \rightarrow \min, \tag{2}$$

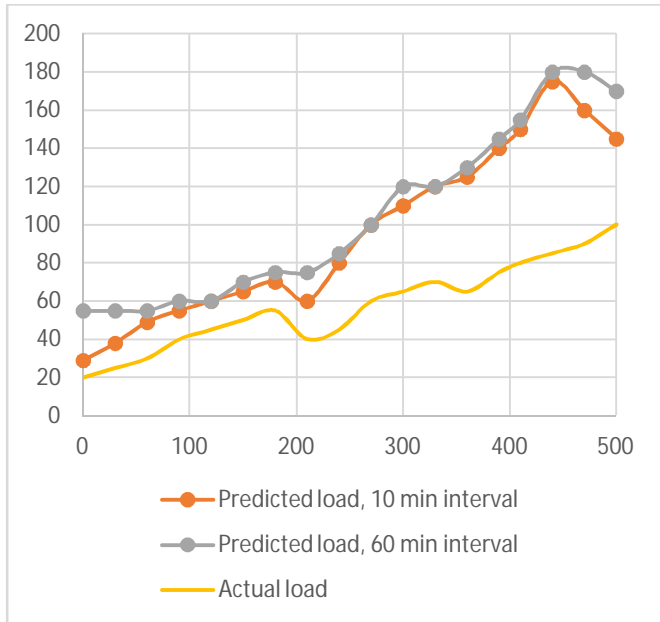
where  $X$  is a time series and  $f(m)$  is a function of the value of the number of changes or intervals  $m$ . The purpose of expression (2) is to minimize the error of load representation and the number of changes. It is a goal function that must be minimized. In some cases, minimizing the square of the number of changes (or any other function of the number of changes) might be preferred. For this work, we set  $f(m) = L \cdot m$ , where  $L$  is the normalization constant.

The objective function helps to minimize the normalized number of changes and the error of representation. The representative error cost function is used to quantify the load representation error. In the most general case, the number of changes and the representative errors can be formulated as utility functions.

The optimal value of the base interval can be determined iteratively. We need to set different values of the number of intervals, also calculate the value of the expression (2) and choose the best minimum with each interval:

$$r_i = \max_{T \in (T_{i-1}, T_i)} X(T). \tag{3}$$

Examples of the representation of time series values are presented in Fig. 2. The error of representation for the case of 10-minute intervals is near 6%, and for the case of 60 minutes, it is almost 19%.



**Figure 2:** Representation of load values depending on different size of adaptation interval

The difference between neighboring representative values should not be less than a certain threshold. Baseline interval values are determined offline and based on long-term statistics (days, weeks, etc.), so it has no effect on the current functioning of the system.

Accordingly, the number of iterations to determine the optimum value of the base interval value may be sufficiently large and selected by the operator depending on the desired accuracy of the solution.

The resource allocation module is called periodically (every window of adaptation or when the threshold is reached) to dynamically divide the resource volume between different network functions running on shared servers on the network.

The adaptation algorithm is started every  $W$  –unit of time. Let  $q_1^0$  denote the length of the queue at the beginning of the adaptation window.

Let  $\lambda_i$  denote the estimate of the intensity of receipt of applications and  $\mu_i$  denote the estimate of the intensity of service in the next window of adaptation (i.e., for the following  $W$  time units).

Then, assuming that the values of  $\lambda_i$  and  $\mu_i$  are constant, and the length of the queue at any point in time  $t$  inside the next window of adaptation is given by:

$$q_i(t) = \max(0; q_i^0 + (\lambda_i - \mu_i)t). \tag{4}$$

The process is modeled as a serving device, that's why the service intensity of the query of the network function is equal to  $\mu_i = \frac{C_i}{S_i}$ , where  $C_i$  – the number of resources of the

network function and  $S_i$  – the average time of service of the query per unit of resource. The average length of the queue during the adaptation window is determined:

$$q_i = \frac{1}{W} \int_0^W q_i(t) dt. \tag{5}$$

The average response time  $T_i$  in the same time interval is estimated:

$$T_i = \frac{q_i + 1}{\mu_i} \tag{6}$$

The parameters of such model depend on its current characteristics. Accordingly, this model is designed to take into account the online response scenarios for dynamic load changes.

The number of resources must be allocated to a network function  $T_i \leq d_i$ . Then the number of resources allocated to the network function  $C_i$  must satisfy next expression:

$$C_i \geq S_i \frac{q_i + 1}{d_i}. \tag{7}$$

Suppose that  $\lambda_{basepred}(t)$  is the baseline predicted flow rate during a certain interval  $t$  obtained from the analysis of previously received data over the past days.

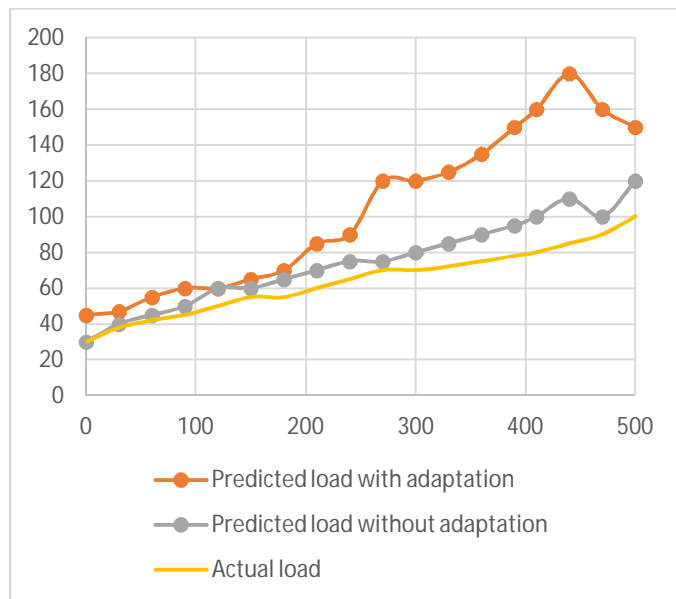
Next, let  $\lambda_{obs}(j)$  be the real intensity of the flow during this interval. The predicted value over the next interval is adjusted using the observed error in accordance with (8):

$$\lambda_{pred}(t) = \lambda_{basepred}(t) + \sum_{j=t-h}^{t-1} \frac{\lambda_{obs}(j) - \lambda_{pred}(j)}{h}. \tag{8}$$

Consider the operation of one block for one day (500 minutes) and consider the given basic component of the estimated intensity of requests for each minute  $\lambda_{basepred}$ , as well as let the average value of the time of service of the application per unit of resource  $s_i$  and it does not change, also assume the availability of a resource of one type.

The simulation results showed (Fig. 3) that the error in the predicted value in comparison with the real one can be 16%, with the "positive" error accounting for 9% of the real intensity of the receipt of applications.

If we do not use the system of dynamic adjustment of the magnitude of the adaptation window and the system of taking into account the previously obtained data in the prediction, the error will be 21%, "positive" error - 13%, from which we can also conclude that the resources will be extremely ineffective.



**Figure 3:** Results of system modeling with the dynamic adaptation of control interval and without them

Thus, the general algorithm can be represented as described below. The base load prediction device first performs a load analysis to identify patterns. On this basis, it is possible to display periods of the clearest patterns.

The base load forecasting device starts periodically (for example, once a day). It divides the previously received data into daily requirements and forecasts the next day's requirement. For load forecasting, we can take the average of the previously received daily load. Complex prediction methods can be used, but this is beyond the scope of this article. As a result of the unit output, we get a forecast of the load for the period.

The controller is responsible for handling the load. The device receives a predicted base load template from a base load forecasting device, estimates and distributes resources needed to handle the load. In particular, the controller uses the performance queuing model described above to determine how many resources will be allocated to meet the requirements for the estimated number of requests without consuming excess capacity.

Expression (7) estimates the number of resources needed to ensure that the target average response time is satisfied. The controller starts every adaptation window or when the actual load exceeds the projected load in order to provide additional resources for overload.

Because the controller does not start when the actual load is lower than predicted, the impact of over-allocation of resources is minimized if the forecast allocation of resources well reflects the basic requirements.

The coordinator sends inbound requests to the server dedicated to load maintenance, and communicates with the controller to provide input load information. It should be noted that the monitoring system monitors the traffic and counts the number of requests.

The monitoring system sets a threshold for the number of requests and a message is sent to the controller if the monitoring system detects an overload. When a controller receives an overload message from the monitoring system, the controller calculates the required number of resources to process the requests properly and dynamically distributes the estimated volume. Then the coordinator redirects the requests and the overload is eliminated.

### 3. CONCLUSION

The simulation results showed that the error in the predicted value compared to the real one can be 16%, with an additional error of 9% of the real intensity of requests.

If we do not use the system of dynamic adjustment of the magnitude of the adaptation window and the system of taking into account the previously obtained data in the prediction, the error will be 21%, the error - 13%, from which we can also conclude that the resources will be extremely inefficient.

### REFERENCES

- [1] Domracheva K.O., Dovzhenko N.M., Dmitrenko V.V. **Analysis of technologies and connection standards for the iot network.** Naukovo-vyrobnychiy zbirnyk «Naukovi zapysky ukrainskoho naukovo-doslidnoho instytutu zviazku». – K.: UNDIZ, 2019. – Vyp 4 (52). – S. 54 – 62. DOI: 10.31673/2518-7678.2019.0305462
- [2] M. Aazam, E.-N. Huh, **Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT**, in: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, IEEE, 2015, pp. 687–694.
- [3] M.M. Rathore, A. Ahmad, A. Paul, G. Jeon, **Efficient graph-oriented smart transportation using internet of things generated big data**, in: 2015 11th International Conference on Signal-image Technology & Internet-based Systems (SITIS), IEEE, 2015, pp. 512–519.
- [4] Melnyk Yurii, Matsko Olexander, Ilin Oleh, Hnidenko Nikolay, Dakova Larisa, Dakov Serhii, Domracheva Kateryna, Dovzhenko Nadiia. **The Process of Network Flows Distribution based on Traffic Engineering Method.** International Journal of Advanced Trends in Computer Science and Engineering. 2019. Volume 8, №6, November – December. Pp. 3036 - 3042
- [5] Syed.Karimunnisa and Dr.Vijaya Sri Kompalli. **Cloud Computing: Review on Recent Research Progress and Issues.** International Journal of Advanced Trends in Computer Science and Engineering. 2019. Volume 8, №2, March – April. Pp. 216 – 223
- [6] Mahdavinejad,M.S., Rezvan,M., Barekatain,M., Adibi,P., Barnaghi,P., Sheth,A.P. **Machine learning for Internet of Things data analysis: A survey**, Journal of Digital Communications and Networks. 2017 vol.4, 161-175. <https://doi.org/10.1016/j.dcan.2017.10.002>
- [7] Ranjan Kumar H S, Ganesh Aithal, Surendra Shetty. **Insights on Effectiveness of Secure Multicast Communication Scheme in**

**Future Wireless Network.** International Journal of Advanced Trends in Computer Science and Engineering. 2019. Volume 8, № 4, July – August. Pp. 1359 - 1367

[8] Koduru Suresh, PVGD Prasad Reddy, Padala Pushkal. **Smart Home Services Using Cloud and Internet of Things.** International Journal of Advanced Trends in Computer Science and Engineering. 2019. Volume 8, № 4, July – August. Pp. 1560 - 1567

[9] Amini, S., Gerostathopoulos, I., &Prehofer, C. (2017). **Big data analytics architecture for real-time traffic control.** 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS). doi:10.1109/mtits.2017.8005605

[10] ELMENDILI, s., el BOUZEKERI, y. and HMINA, N. (2019). **Big Data Processing Platform for Smart City,** IEEE - International Symposium on Advanced Electrical and Communication Technologies (ISAECT2018), 21-23 november, 2018, Morocco IEEE Conference Publication. [online] Ieeexplore.ieee.org. Available at: <https://ieeexplore.ieee.org/document/8618812> [Accessed 2019]. <https://doi.org/10.1109/ISAECT.2018.8618812>

[11] D. Kelaidonis, A. Somov, V. Foteinos, G. Poulis, Stavroulaki, P. Vlacheas, P. Demestichas, A. Baranov, Biswas, and R. Giaffreda, “**Virtualization and cognitive management of real world objects in the internet of things,**” in Green Computing and Communications (GreenCom), 2012 IEEE International Conference on. IEEE, 2012, pp. 187– 194. <https://doi.org/10.1109/GreenCom.2012.37>