UDC621.37

RESEARCH OF ALGORITHMS OF SERVICE OF LOADINGS IN FOG – NETWORKS

MoshynskaA.V., SozonnikG.D., MaltsevA.G.
Institute of Telecommunication Systems,
Igor Sikorsky Kyiv Polytechnic Institute, Ukraine
E-mail: maltsev.andrew@gmail.com; avmoshinskaya@gmail.com

ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБСЛУГОВУВАННЯ НАВАНТАЖЕНЬ У FOG – МЕРЕЖАХ

У роботі представлено основні принципи забезпечення автоматизованої роботи та оптимізації бездротових мереж при використанні самооптимізуючих систем (SON) за допомогою туманних мереж.

Wireless networks are and will continue to be a fundamental cornerstone of the global digital economy and our connected society, provisioning diverse services for people and things. The proliferation of smart user devices, such as smart phones, laptops, and tablets, is pushing the current wireless networks to their limits. Wireless networks are experiencing an unprecedented traffic growth with an estimated compound annual growth rate of 0.6-1.0 and an increasing variety of services and applications, each with potentially different traffic patterns and quality of service (QoS) and quality of experience (QoE) requirements, for example, ultra- high data rate and/or reliability, and ultra-low latency. To cope with the continuing traffic growth and service expanding, future wireless networks will have to be heterogeneous and densely deployed, featuring the coexistence of different radio access technologies (RATs), such as LTE/LTE-advanced, Wi-Fi, loT, 5G NR, etc.Wireless network self-optimization comprises various mechanisms that optimize network parameters during operation according to measurement data taken at different parts of the network.

Fog computing, also called Edge Computing, is intended for distributed computing where numerous "peripheral" devices connect to a cloud. (The word "fog" suggests a cloud's periphery or edge). Many of these devices will generate voluminous raw data (e.g., from sensors), and rather than forward all this data to cloud-based servers to be processed, the idea behind fog computing is to do as much processing as possible using computing units co-located with the data-generating devices, so that processed rather than raw data is forwarded, and bandwidth requirements are reduced. An additional benefit is that the processed data is most likely to be needed by the same devices that generated the data, so that by processing locally rather than remotely, the latency between input and response is minimized.

A self-organizing network (SON) is an automation technology designed to make the planning, configuration, management, optimization and healing of mobile radio access networks simpler and faster. SON functionality and behavior has been defined and specified in generally accepted mobile industry recommendations produced by organizations such as 3GPP (3rd Generation Partnership Project) and the NGMN (Next Generation Mobile Networks).

Nevertheless, most existing SON solutions are mainly based on heuristics, with the automated information processing limited to relatively simple methods [2]. Many open challenges of SON remain unsolved, for example, the automated coordination between different SON functions, and the trade-off between centralized and distributed SON implementations.

Fog computing can be used to enhance the operations, administration, and maintenance (OAM) processes for distributed MLB in support of multi-RAT coordination.

Wireless networks are and will continue to be a fundamental cornerstone of the global digital economy and our connected society, provisioning diverse services for people and things.[1] Different network architectures have been considered for the realization of SON, including centralized SON, distributed SON, and hybrid SON. The choice of the SON architecture may affect the performance and/or efficiency of the SON functionalities for wireless networks.

In centralized SON (C-SON), the self-optimizing algorithms mainly reside in the network management system (NMS). In distributed SON (D-SON), the SON functions are distributed across the edge of the network, typically in BSs and APs.

Hybrid SON (H-SON) combines the centralized network management and the distributed SON functionalities.

Mobility load balancing (MLB) is a self-optimization function that aims to maximize the network capacity (or to optimize user QoS or QoE) through optimizing the distribution of user traffic load across the network nodes and radio resources. As a result, traffic overload or congestion at any network node can be avoided. MLB is usually implemented in a distributed manner and relies on the traffic load estimation and radio resource utilization status exchanged among neighboring network nodes via the X2 interface.

For last several years wireless networks used offloading to edge servers. The main benefit of offloading to edge servers is that it can support real-time tasks by providing low latency connectivity for the application on mobile devices. The problem arises when there are a number of mobile devices in hot spots. In such a scenario, multiple devices try to access nearby edge servers, which have limited capabilities compared to the central cloud server. When a load balancing technique is not applied, some edge servers are prone to failures. However, implementing a load balancing technique for edge servers in edge cloud computing environments is not an easy task, and a few load balancing techniques are implemented for edge servers; previous load balancing techniques are mainly for virtual machine or container consolidation Electronics 2020, 9, 686 3 of 13 and orchestration in the central cloud server. Thus, we design and implement a load balancing technique in edge cloud computing environments with graph coloring based on a genetic algorithm.

But now, using FOG – computing there are much more ways of offloading. Such as Genetic Algorithms.

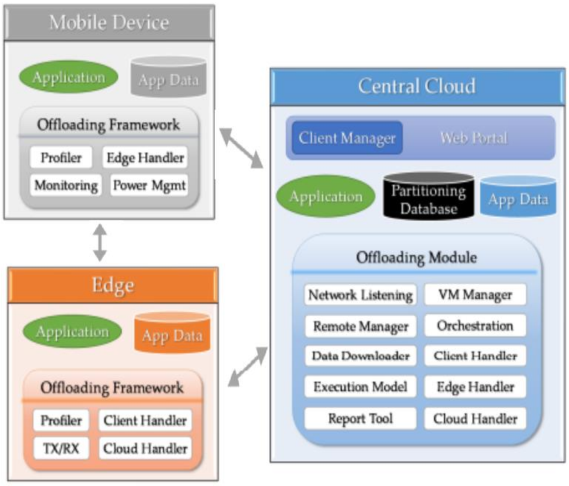

Fig.1. An edge cloud architecture and its interaction.

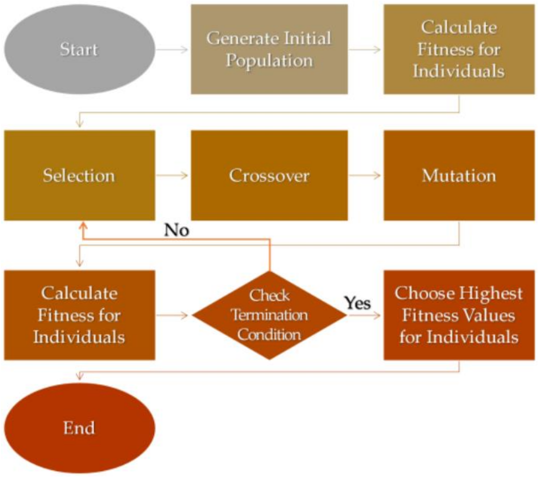

Fig.2. The process of a genetic algorithm.

To mimic natural evolution processes, it reflects the natural selection process with a fitness function for individuals and produces the next generation based on crossover and mutation processes. Figure 3 shows the process of a genetic algorithm. It starts with the initial population of the genesis generation and uses of a genetic algorithm. It starts with the initial population of the genesis generation and uses a fitness function to measure scores for individuals in the population. Then, it selects a target based on the fitness function and process the crossover and mutation processes to mimic the evolutionary process of humans.

Specifically, a genetic algorithm can be used to solve optimization problems for large datasets and search space.

Conclusions

The advantages of employing a genetic algorithm are as follows.

Firstly, it can solve fundamental problems with global maxima and minima.

Secondly, it can handle complex mathematical representations with low complexity, while the linear programming approach exhibits high complexity.

Thirdly, it supports problems that are difficult to represent in mathematical forms. Fourthly, it is resilient to data noise and failures.

Fifthly, it is easy to implement solutions by exploiting parallel and distributed processing.

Lastly, it can learn with new data and information in a continuous fashion to achieve global maxima and minima.

References

1.Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the Internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing, Helsinki, August 2012, pp 13-16.

2.Lopez-Perez D, Guvenc I, Chu X, Mobility management challenges in 3GPP heterogeneous networks. IEEE Commun Mag, 2012, pp 70-78.

3.Mao Y, Zhang J, Letaief KB, Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J Sei Areas Commun, 2016, pp 3590-3605.