Smart City Insights

www.sci.reapress.com

Smart City Ins. Vol. 1, No. 1 (2024) 43–50.

Paper Type: Original Article

A Review on Load Balancing Evolution in Cloud Computing

Saswat Misra*

Department of Computer Engineering, KIIT AS University, Bhubaneshwar 751024, Odisha, India; 21052702@kiit.ac.in.

Citation:

Received: 26 June 2023 *Revised: 9 August 2023* *Accepted: 12 January 2024*	Misra, S. (2024). A review on load balancing evolution in cloud computing. *Smart City Insights, 1 (1), 43-50.*

Abstract

Cloud computing is basically an on-demand service provider that provides services on a large scale. To provide services on a large scale, a large number of servers are interconnected with each other. As technology evolves, providing a smooth and efficient transfer of data and services is necessary. It is where a load balancer comes into play; it distributes the incoming traffic amongst multiple other servers to offer a smooth and efficient transfer of services. But there is a downside to load balancing. Although it provides a smooth transfer of data and resources, it becomes a challenge due to the geographical conditions of widely spread data across the globe. This paper will thoroughly explore the advantages and disadvantages of various load-balancing strategies, providing a comprehensive understanding of how load balancing operates. By the end of this paper, we'll also conclude whether using a load balancer will help us in cloud computing.

Keywords: Cloud computing, Load balancing, Load balancing strategies.

1 | Introduction

Cloud computing has become a popular buzzword for different technologies, services, and concepts. It is a network technology that provides services to various customers [1]. It provides both hardware and software applications along with software development platforms.

Some types of services provided by cloud platforms include Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). These services are often provided by companies like Amazon, Microsoft, IBM, SAP, Oracle, Google, VMware, Salesforce and others [2]. An overview of Access to cloud computing shown in *Fig. 1*.

Corresponding Author: 21052702@kiit.ac.in

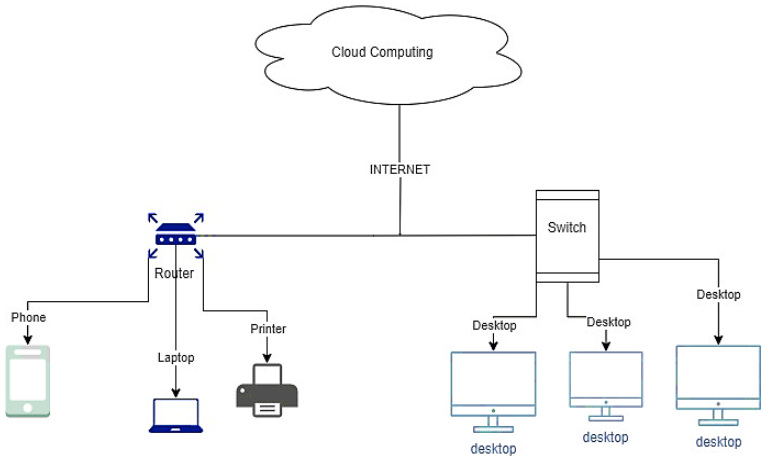


Fig. 1. Access to cloud computing.

These services help clients get on-demand service for a specific resource that they want to use. These features made cloud computing more popular and significantly increased demand for cloud services. It led to more processes being carried out in the cloud, increasing the load on the cloud service provider's servers. To overcome this issue, a Load balancer was introduced. Load balancing is distributing network traffic equally across a pool of resources that support an application [3]. The diagram below is a representation of a load balancer. An overview of load balancing in cloud computing shown in *Fig. 2*.

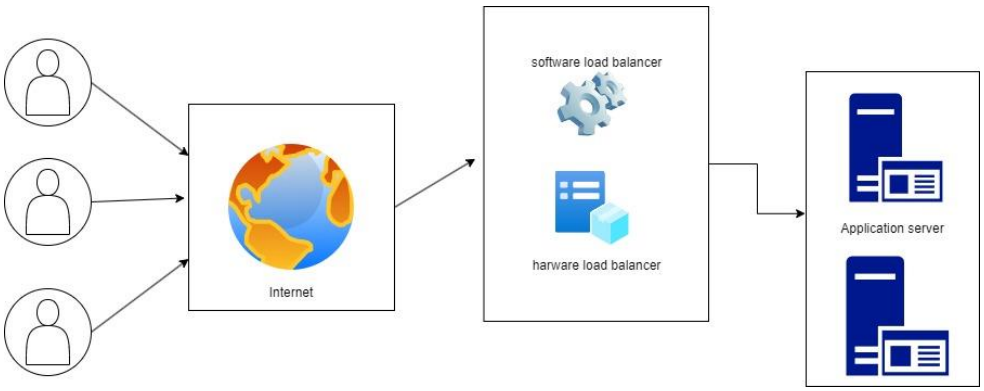


Fig. 2. load balancing in cloud computing.

However, there were several challenges in load balancing, such as complexity, geographical distance, latency, and misconfiguration. Therefore, different types of load balancing were introduced.

I. Application Load Balancing: Application load balancers play a vital role in ensuring the smooth functioning of complex modern applications. These load balancers optimize performance and prevent system overload by appropriately distributing incoming traffic across multiple servers. Understanding the various algorithms and techniques load balancers use can help you make informed decisions when implementing them in your environment [4].

II. Network Load Balancing: They examine IP addresses and other network information to redirect traffic optimally. They track the source of application traffic and can static IP addresses to several servers [5].

III. Global server load balancing: GSLB or Global Server Load Balancing is a practice of distributing internet traffic amongst a large number of connected servers dispersed around the world.

IV. DNS load balancing: In this, we can configure our domain and route network requests across the pool of resources on our domain.

2 | Literature Review

Many recent approaches aim to improve the performance of cloud computing by providing efficient load-balancing techniques. You can see the load balancing classification shown in *Fig. 3*.

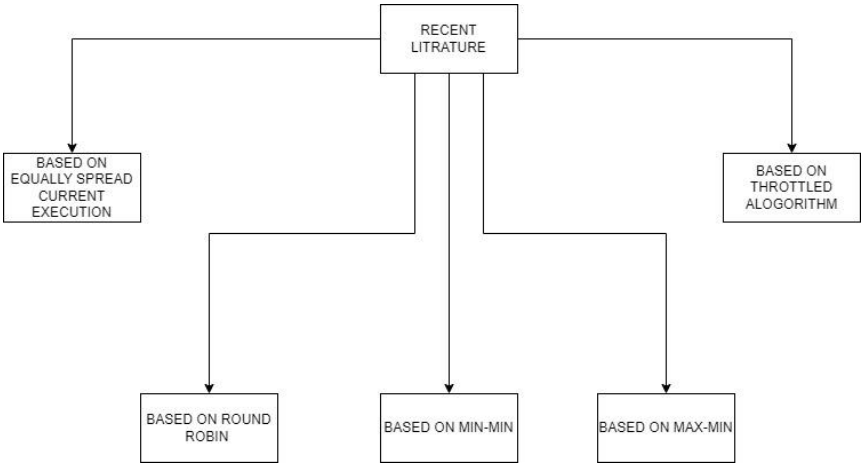
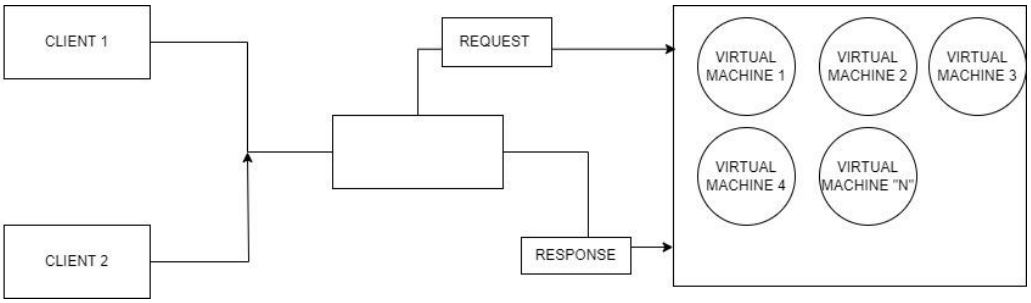


Fig. 3. load balancing classification.

Load balancing based on throttled algorithm

The throttled algorithm is also known as the dynamic LB algorithm. This load balancer aims to search for a suitable virtual machine that can respond effectively and perform tasks efficiently upon receiving a request from the client [6]. It keeps a list of all the virtual Machines along with their index value, which is stored in an allocation table; this table shows the virtual machine's state, i.e., if they are available, busy or idle. If a virtual machine is available and has space, then the task is allocated to that machine; if not available and a virtual machine is found, the request is queued for fast processing. It performs better than round-robin but doesn't consider advanced requirements for load balancing. The *Fig. 4* has a proper explanation for it [7].

Fig. 4. load balancing handling request.



Advantages of throttled load balancing

I. Throttled load balancing helps in optimizing load balancing by ensuring that every server operates within its limits.

II. It improves the performance by preventing a server from becoming overloaded with requests.

III. It can easily adapt to the changes in the workload environment.

Disadvantages of throttled load balancing

I. Although throttled load balancing improves performance, it is a complex algorithm, especially in large-scale distributed systems.

II. This algorithm consumes the computational resources that may impact the overall system performance.

III. Although this algorithm aims to distribute the incoming requests, there might be latency issues, especially if there are network delays.

Load balancing based on equally spread current execution

In this type of dynamic algorithm, the job's size is considered the priority; then, it distributes the workload to a virtual machine with a lighter load. This technique, also known as the Spread Spectrum technique, spreads the workload to different nodes [8]. This algorithm uses a queue to store the request and distribute the workload to various virtual machines. It has a major drawback: it could overheat when updating the index due to the communication between the load balancer and the data centre. The process has been further explained in the *Fig. 5*.

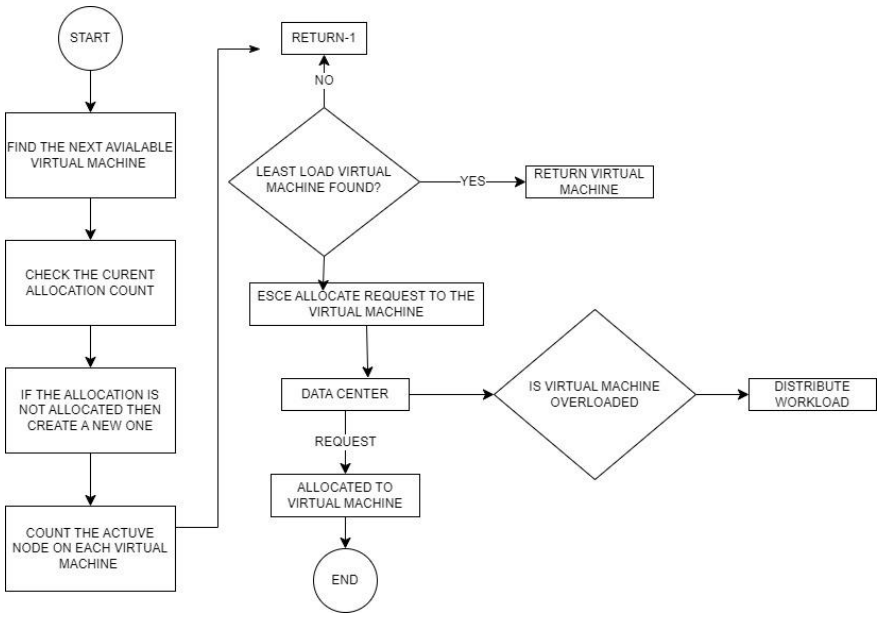


Fig. 5. Load balancing working flowchart.

Advantages of equally spread current execution

I. This algorithm ensures that resources are evenly distributed among servers or processing units, which helps to prevent individual servers from becoming overloaded.

II. It is highly saleable. It can easily scale by adding or removing servers or processing units.

III. It helps in fault tolerance by distributing the workload across multiple servers. If one server fails or becomes unavailable, the remaining servers can continue to handle the workload without being overwhelmed, ensuring continuous availability of services [9].

Disadvantages of equally spread current execution

I. This algorithm can lead to overhead in monitoring, coordination, or communication, which can consume computational resources and impact overall system performance.

II. In some cases, it can lead to underutilization, and this can lead to the workload not being able to be distributed properly.

III. It can't adapt to this fast-paced workload environment; it struggles with the spikes in the workload environment.

Load balancing using round robin

This algorithm works in a circular and ordered procedure where each process is assigned a fixed time slot without priority. This algorithm is very commonly used due to its simple implementation. A common problem in this load-balancing technique is that after user requests, the allocation state of that virtual machine isn't saved or updated, as seen in the *Fig. 6*. [10].

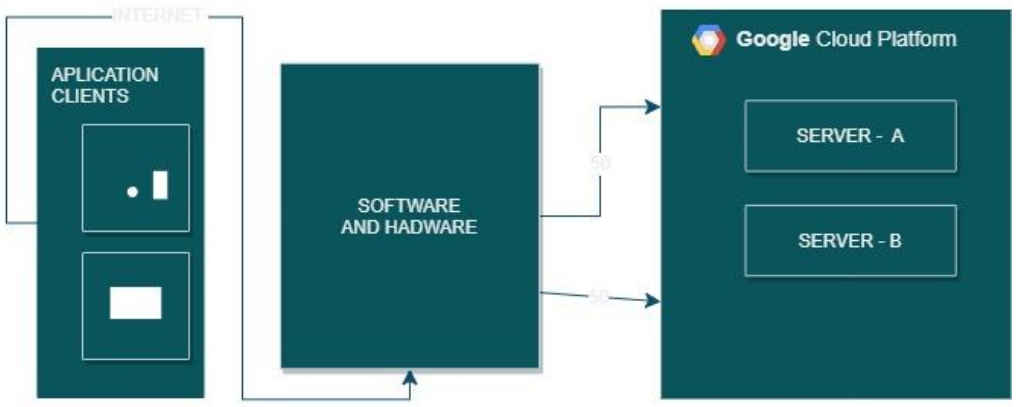


Fig. 6. Hardware and software in load balancing.

Advantages of round robin in load balancing

I. Round-robin load balancing is easy to use and implement; it doesn't require any complex algorithm.

II. It ensures that each server receives an equal share of incoming requests.

III. It is highly saleable. It can easily scale up and down according to the number of servers.

Disadvantages of round-robin in load balancing

I. It differentiates between server capacities and processing capabilities. As a result, servers with higher capacities may be underutilized, while servers with lower capacities may become overloaded.

II. It doesn't consider the server's health status, so that it may send requests to servers facing an issue.

Load balancing using the min-min algorithm

In this algorithm that shown in *Fig. 7*, the minimum completion time is considered for scheduling. it has a lot of shortcomings, like the inability to run tasks simultaneously, the algorithm gives high priority to the smaller tasks, which leads to longer waiting time for the larger tasks, resulting in an imbalanced virtual machine load [11].

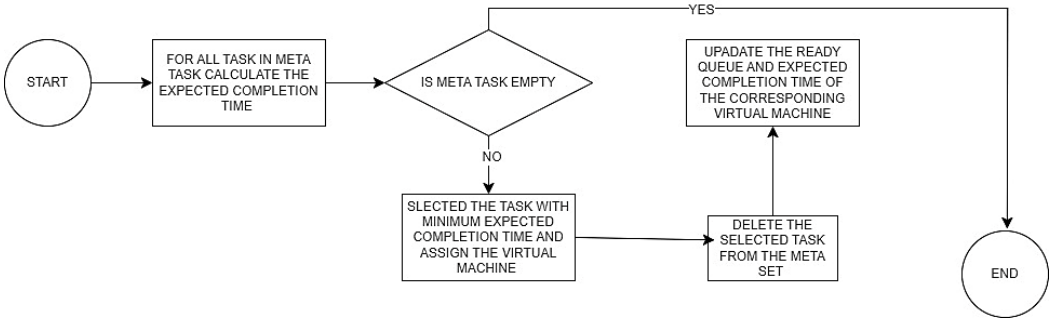


Fig. 7. Load balancing in min-min.

Advantages of min-min in load balancing

I. The primary advantage of the min-min algorithm is that it requires a minimum period to complete all tasks together.

II. It effectively allocates the tasks based on their execution time, which helps in maintaining each resource active and productive.

III. It is both flexible to workload and adaptive.

The disadvantage of min-min in load balancing

I. Despite minimizing the makespan, it may not always produce an optimal schedule, especially with complex task dependencies.

II. It relies heavily on accurate estimations of task execution times to make scheduling decisions. Inaccurate estimations can lead to delays or performance degradation, particularly if tasks take longer to execute than initially estimated.

Load balancing using max-min algorithm

This algorithm is the same as min-min, but instead, high priority is given to the machine that can perform various tasks in maximum time. Then, from the selected task, the task that requires maximum time is executed again, and then the task with minimum time will be executed. The *Fig. 7* is a representation of the max-min algorithm [12].

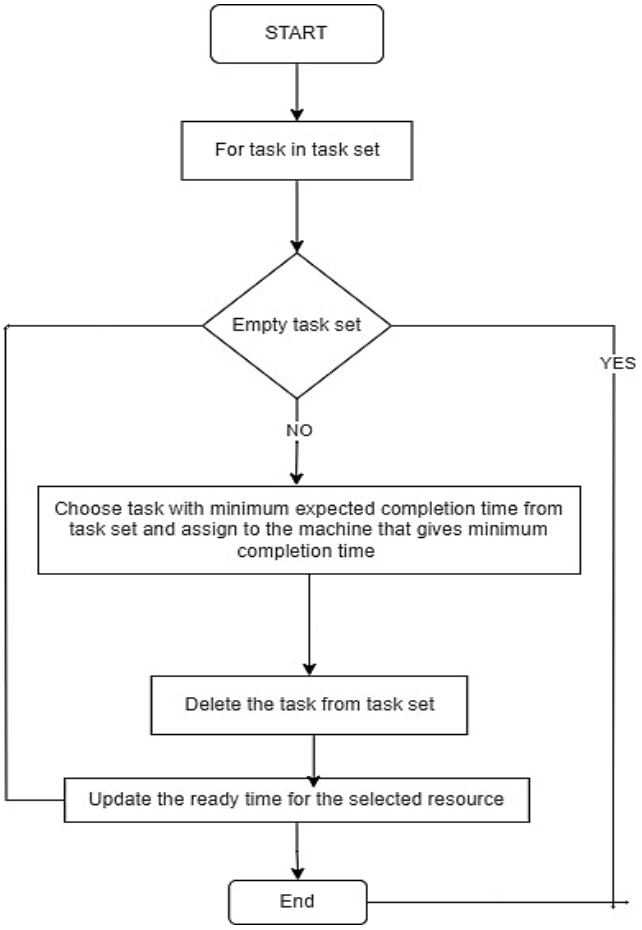


Fig. 7. Load balancing in max-min.

Advantages of max-min in load balancing

The primary advantage is its ability to maximize the minimum completion time of all tasks.

I. It prioritizes tasks with longer execution times over shorter ones.

II. It promotes load balancing by distributing tasks amongst all available resources based on their execution time.

The disadvantage of max-min in load balancing

I. In some cases, prioritizing tasks with longer execution times may lead to sub-optimal resource utilization or delays in completing the tasks.

II. Just like the min-min algorithm, max-min also relies on accurate estimations of task execution to make scheduling decisions.

Challenges associated with load balancing

Cloud computing depends on the proper utilization of resources, and load-balancing techniques are responsible for smoothly providing resources and services to clients. Yet there are various challenges associated with it.

Unpredictable and dynamic workload: distributed systems often experience dynamic workloads with unknown levels of incoming network traffic. Load balancers must adapt to these to function properly in real time.

Heterogeneous environments: a distributed system consists of a heterogeneous environment that might have resources with different capacities and capabilities. The load balancer must effectively distribute workload across varying resources.

Complexity of algorithm: the algorithm should be simple, effective, and easy to use because it may decrease the efficiency and performance of cloud computing.

Security: due to load balancers working in an environment with multiple server traffic. Load balancer must improve their security so that the data is not lost and to protect the data and resources from cyber-attacks.

Proposed work

Considering the advantages and disadvantages of each load-balancing algorithm, this paper aims to address the limitations while leveraging the strengths to enhance overall performance and efficiency in diverse cloud computing environments.

Optimizing distribution algorithm in throttled load balancing

I. Implementing more sophisticated algorithms like weighted round robin or connection to ensure balanced resource utilization.

II. Monitor continuously and analyze server loads to adapt to workload patterns.

Efficient task scheduling in equally spread load balancing

Using advanced task scheduling techniques such as priority-based scheduling to ensure that critical or time-sensitive tasks receive higher priority and are allocated resources accordingly.

Health monitoring and failure detection in round robin

Integrating health monitoring and failure detection mechanisms, periodically checking the status of servers, and removing unhealthy or failed servers. It prevents round-robin from sending requests to non-responsive or malfunctioning servers, improving overall system reliability.

Dynamic task rescheduling in min-min and max-min

Implementing a mechanism for dynamic task rescheduling that'll check the execution status of tasks and adjust their allocation based on real-time information. It will allow the system to adapt to the changing conditions and will prioritize the critical tasks.

3 | Conclusion

In conclusion, this paper comprehensively studies various load-balancing strategies and algorithms in cloud computing environments. With the help of analysis of different load balancing techniques and algorithms and listing their pros and cons, we have gained great insight into their potentials, applications, limitations and ways to improve. This paper aims to improve and optimize the existing algorithms and enhance their adaptability,

scalability, and security so that these algorithms can be used to reach their potential and meet the requirements of modern digital environments. This study is a foundation for further research and development in this field. By addressing the identified challenges and implementing proposed enhancements, we can use the full potential of cloud computing technologies for future applications.

Funding

This research received no external funding.

Data Availability

All the data are available in this paper.

Conflicts of Interest

The author declare no conflict of interest.

References

[1] Pham, X. Q., Nguyen, T. D., Huynh, T., Huh, E.-N., & Kim, D.-S. (2023). Distributed cloud computing: architecture, enabling technologies, and open challenges. *IEEE consumer electronics magazine*, *12*(3), 98–106. DOI:10.1109/MCE.2022.3192132

[2] Mohapatra, H. (2021). Socio-technical challenges in the implementation of smart city. *2021 international conference on innovation and intelligence for informatics, computing, and technologies (3ICT)* (pp. 57–62). IEEE. DOI: 10.1109/3ICT53449.2021.9581905

[3] Cui, Z., Cui, P., Hu, Y., Lan, J., Dong, F., Gu, Y., & Hou, S. (2021). Closer: scalable load balancing mechanism for cloud datacenters. *China communications*, *18*(4), 198–212. DOI:10.23919/JCC.2021.04.015

[4] Mohapatra, H., & Rath, A. K. (2020). Nub less sensor based smart water tap for preventing water loss at public stand posts. *2020 IEEE microwave theory and techniques in wireless communications (MTTW)* (Vol. 1, pp. 145–150). IEEE. DOI: 10.1109/MTTW51045.2020.9244926

[5] Kalafatidis, S., & Mamatas, L. (2022). Microservices-adaptive software-defined load balancing for 5G and beyond ecosystems. *IEEE network*, *36*(6), 46–53. DOI:10.1109/MNET.004.2100333

[6] Mohapatra, H., & Rath, A. K. (2020). Fault-tolerant mechanism for wireless sensor network. *IET wireless sensor systems*, *10*(1), 23–30. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-wss.2019.0106

[7] Rehman, M. A. U., ud din, M., Mastorakis, S., & Kim, B.-S. (2023). Foggyedge: an information-centric computation offloading and management framework for edge-based vehicular fog computing. *IEEE intelligent transportation systems magazine*, *15*(5), 78–90. DOI:10.1109/MITS.2023.3268046

[8] Mohapatra, H., & Rath, A. K. (2019). Fault tolerance in WSN through PE-LEACH protocol. *IET wireless sensor systems*, *9*(6), 358–365. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-wss.2018.5229

[9] Xie, R., Tang, Q., Qiao, S., Zhu, H., Yu, F. R., & Huang, T. (2021). When Serverless computing meets edge computing: architecture, challenges, and open issues. *IEEE wireless communications*, *28*(5), 126–133. DOI:10.1109/MWC.001.2000466

[10] Mohapatra, H., & Rath, A. K. (2019). Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT. *IET wireless sensor systems*, *9*(6), 447–457. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-wss.2019.0081

[11] Chen, L., DeJana, R., & Nassar, T. (2021). Sharing enterprise cloud securely at IBM. *IT professional*, *23*(1), 67–71. DOI:10.1109/MITP.2020.2977029

[12] Mohapatra, H., & Rath, A. K. (2020). Survey on fault tolerance-based clustering evolution in WSN. *IET networks*, *9*(4), 145–155. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-net.2019.0155