INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(4), 459-474, August 2019.

ESBL: Design and Implement A Cloud Integrated Framework for IoT Load Balancing

G. Balakrishna, M.N. Rao

Gubba Balakrishna
Koneru Lakshmaiah EducationFoundation
Department of Computer Science and Engineering
me2balu@gmail.com

Nageswara Rao Moparthi
Koneru Lakshmaiah Education Foundation
Department of Computer Science and Engineering
rao1974@gmail.com

Abstract: The continuous growth in wireless communication, the demand for sophisticated, simple and low-cost solutions are also increasing. The demand motivated the researchers to indulge into inventing suitable network solutions ranging from wireless sensor networks to wireless ad-hoc networks to Internet of Things (IoT). With the inventions coming from the researchers, the demand for further improvements into the existing researchers have also growth upbound. Initially the network protocols were the demand for research and further improvements. Nevertheless, the IoT devices are started getting used in various fields and started gathering a huge volume of data using complex application. This invites the demands for research on load balancing for IoT networks. Several research attempts were made to overcome the communication overheads caused by the heavy loads on the IoT networks. Theses research attempts proposed to manage the loads in the network by equally distributing the loads among the IoT nodes. Nonetheless, in the due course of time, the practitioners have decided to move the data collected by the IoT nodes and the applications processing those data in to the cloud. Hence, the challenge is to build an algorithm for cloud-based load balancer matching with the demands from the IoT network protocols. Hence, this work proposes a novel algorithm for managing the loads on cloud integrated IoT network frameworks. The proposed algorithm utilizes the analytics of loads on cloud computing environments driven by the physical host machines and the virtual environments. The major challenge addressed by this work is to design a load balancer considering the low availability of the energy and computational capabilities of IoT nodes but with the objective to improve the response time of the IoT network. The proposed algorithm for load balancer is designed considering the low effort integrations with existing IoT framework for making the wireless communication world a better place.

Keywords: IoT, cloud integrated load balancer, inter quartile correlation, static threshold utilization.

1 Introduction

The most recent ground-breaking innovation in the field of networking is Internet of Things (IoT). The IoT networks have gained a huge popularity for making the communication protocols working with three simple components as gather the data from the open operational environment using sensors, the independent machine to machine communication and the cloud computing environment for the applications and the data collected by the sensor agents. An important aspect of IoT is their protection against cyber attacks [1]. The increasing popularity of smart conceptual factors of modern appliances and life style made IoT an integral part of the modern wireless communications. IoT frameworks are frequently considered as a layered secluded engineering of a computerized innovation. The gadget layer alludes to the physical parts: CPS, sensors or machines. The system layer comprises of physical system transports, distributed computing and correspondence conventions that total and transport the information to the administration layer [3], which comprises of uses that control and consolidate information into data that can be shown on the driver dashboard. The best most stratum of the stack is the substance layer or the UI.

The historical backdrop of the IoT starts with the development of the programmable rationale controller by Dick Morley in 1968, which was utilized by GM in their programmed transmission producing division. These PLCs took into consideration fine control of individual components in the assembling chain. In 1975, Honeywell and Yokogawa presented the world's first DCSs, the TDC 2000 and the CENTUM, independently. These DCSs were the following stage in permitting adaptable process control all through a plant, with the additional advantage of reinforcement redundancies by conveying control over the whole framework, disposing of a solitary purpose of disappointment in a focal control room.

Nonetheless, with the increasing demand for further improvements on IoT load balancing for cloud integrated applications and frameworks this work proposes a novel load balancing algorithm.

The rest of the work is furnished such as in the section 2, the routing demands for any IoT network is analysed, in section 3, the benefits of cloud computing service models are analysed, in section 4, the outcomes from the parallel researches are analysed, in the section 5, the novel proposed load balancing algorithm is presented, in the section 6, the integration possibilities of the proposed algorithm into any existing IoT framework is discussed, in the section 7, the results obtained from the proposed framework is discussed and analysed, in the section 8, the comparative analysis with the standard protocols are carried out to establish the significance of improvements and this work presents the final conclusion of the research in the section 9.

2 IoT routing strategies demands

In this section of the work, the fundamental demands for routing strategies on IoT is elaborated and discussed to realise the break points for performance improvements using load balancing.

Some expansive scale remote information securing, and activation related applications utilize low-fuelled installed gadgets. These applications incorporate accuracy horticulture, building the board/mechanical robotization, vehicular specially appointed systems, and urban systems/vitality and water lattices to fabricate more intelligent urban communities. In these remote sensors organizes, the installed gadgets work under serious vitality limitations, which results in calculation, stockpiling, and radio-transmission related imperatives [Fig. 1]. They likewise impart over a lossy channel.

Traffic examples and information stream inside LLN are profoundly directional. The examples can be characterized as multipoint-to-point traffic, point-to-multipoint traffic, or point-to-point traffic. In MP2P traffic, for instance, detected data from various detecting hubs is steered to an Internet application by means of LBR. P2MP traffic is seen when an inquiry asks for is produced using the Internet and steered by means of the LBRs and LLN switches to various field hubs. P2P traffic happens when control data should be sent to actuator or ready data is gotten from a sensor.

The RFC reports depict the key usefulness and steering prerequisites for urban LLNs:

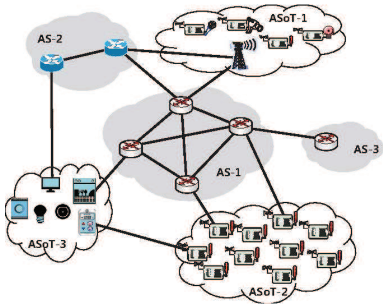

Figure 1: IoT routing building blocks

2.1 Arrangement of nodes

In a run of the mill urban system organization, hundreds or thousands of nodes with pre-customized functionalities are taken off. The steering convention should represent these variables and bolster self-association and self-design at the most reduced conceivable vitality cost.

2.2 Affiliation and disassociation of nodes

After the instatement stage, nodes may join or leave the system at self-assertive occasions. The steering convention likewise ought to most likely handle circumstances where a breaking down node may influence or endanger the general directing effectiveness.

2.3 Ordinary estimation detailing

The information directing calculation and choice may rely upon the detected information, the recurrence of revealing, the measure of vitality staying in the nodes, the reviving example of the vitality searched nodes, or different variables.

2.4 Scalability

The directing convention must almost certainly bolster a field organization of a couple of hundred to a huge number of sensor nodes without decaying chosen execution parameters beneath configurable edges.

Hence, it is natural to understand that the scalability is one of the key factors for low power consuming devices, such as IoT devices. Thus, the scalability of the IoT network can be improved using cloud-based load balancing strategies.

Thus, in the next section of the work, the cloud computing service types with related advantages are discussed.

3 Cloud computing services

This work proposes a novel framework to balance the IoT network data processing loads using the cloud computing services. Hence, it is a prime importance to realize the cloud computing service types and identify the benefits, which can be utilized for balancing IoT network loads.

3.1 Using cloud software capabilities

The ability gave to the data consumers is to utilize applications running on a cloud foundation. The applications are available from different gadgets through either a thin interface, for

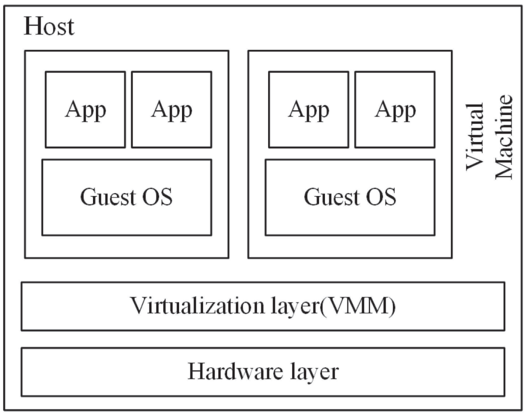

Figure 2: Cloud provisioning capabilities using VM

example, an internet browser, or a programable interface. The consumer does not oversee or control the hidden cloud framework including system, servers, working frameworks, stockpiling, or even individual application capacities, with the conceivable exemption of constrained client explicit application design settings.

3.2 Using cloud operational capabilities

The ability gave to the application consumers to send application onto the cloud framework, data centre owners made or gained applications made utilizing programming dialects, libraries, administrations, and apparatuses bolstered by the supplier. The application owner does not oversee or control the basic cloud framework including system, servers, working frameworks, or capacity, however has power over the conveyed applications and potentially design settings for the application-facilitating condition.

3.3 Using cloud provisioning capabilities

The capacity gave to the application owners to arrangement handling, stockpiling, systems, and other key registering assets where the application consumers can send and run self-assertive programming, which can incorporate working frameworks and applications. The application owner does not oversee or control the basic cloud foundation but rather has command over working frameworks, stockpiling, and conveyed applications; and potentially constrained control of select systems administration parts

Henceforward, it is natural to realize that the provisioning capabilities of cloud computing can be utilized to handle the loads on applications running and generating data from any IoT network.

The provisioning of resources on cloud can be achieved using virtual machines [Figure 2]. Virtual machines are operating frameworks or application situations that is introduced on programming, which emulates devoted equipment. The end client has indistinguishable experience on a virtual machine from they would have on committed equipment. Specific programming, called a hypervisor, copies the PC customer or server's CPU, memory, hard circle, arrange, and other equipment assets totally, empowering virtual machines to share the assets. The hypervisor can imitate numerous virtual equipment stages that are disengaged from one another, enabling virtual machines to run Linux and Windows Server operating frameworks on the equivalent fundamental physical host. Virtualization limits cost by lessening the requirement for physical equipment frameworks.

4 Outcomes from the parallel researches

The origination of the modern research on IoT started with the newer directions explored by the S. Oteafy et al. in [15]. The work [15] defines the possibilities of utilizing the existing frameworks and strategies from wireless sensor networks. Though, this work formulated the foundation of the research, the deficiencies identified in this work is the designed networks are highly generalized and cannot be extended to the modern day demands from IoT networks. Soon the demand for specified routing strategies are identified and Q. Le et al. in [9] have proposed the enhanced routing strategies for WSN matching with the demands of IoT networks. The shortcoming from this proposed algorithm was the missing component of balancing load. This shortcoming was restructured by the C. Petrioli et al. in [17].

With the initiations by C. Petrioli et al. [17], the newer challenges are identified, and multiple parallel research outcomes are also demonstrated. The work by J. Guo et al. [6] for managing the large-scale network and the work by H. Y. Kim et al. [8] for increasing the life time of the network have demonstrated significant improvements over the existing situations.

Yet another challenge of the present IoT networks is the IoT nodes are deployed for various purposes and must collected information from various sources, in various formats and in various time durations. Hence, managing the load for higher to lower data ranges is obviously a challenge and must be addressed. The work of S. M. A. Oteafy et al. [14] demonstrates the possibilities for self-adaptive or self-adjusting algorithm for balancing loads. The IoT frameworks are based on the standard network protocols and the standard network stack, thus for balancing loads, the capabilities can be utilized. This ideology was proposed by Di Marco et al. in [4] for managing the loads using the network stack functionalities.

During the routing of the data, load balancing being the prime concern, the energy efficiency is also an important factor. With the improvements of the routing algorithms, it is been observed that, the higher complexities of the routing algorithms are pooling a lot of energy for execution. The energy consumptions must be reduced in order to increase the life span of the IoT network.S. A. Alvi et al. in [2] proposes a guideline for framing the routing algorithms following the green computing policies.

Hence, with different recommendations from various researchers on various aspects of IoT frameworks, there was a significant demand to summarize concrete guidelines for IoT design, deployment and managements. The recommendations were well summarized by M. Wu et al. in their work [19].

In spite of number of several prominent outcomes from the research community on IoT, it was a great difficulty to influence the practitioners to use the IoT frameworks as the initial implementations are costly for changing into a newer dimension. After the work of D. Evans et [5] explaining the benefits of using IoT networks for various demands, a high amount of adaptations was observed. Within few months, the demand for IoT implementation was increased to a greater extend. Naturally, the research community was interested to find the details of implementation as to what extend the innovations were helping the practitioners. The survey on the findings was reported timely by L. Mainetti et al. in [12].

The report by L. Mainetti et al. [12] have uncovered a newer research problem for the research community. The IoT network frameworks were designrf to sustain for shorter period and cannot be maintained for a longer duration. Thus, the practitioners have raised a serious concern about the stability of the networks as further improvements or implementation can increase the cost to a greater extend. The solution to this problem was proposed by S. A. Karthikey et al. [7] for making the IoT networks fault tolerant and cost effective.

Further, the resource management schemes are also to be made economical. Many of the parallel research attempts have demonstrated the best use of economic planner to drive the resource management as demonstrated by T. Menouer et al. [13]

Building a resource clusters on cloud for IoT based networks also proven to be significant for load optimization as suggested by R. Peinl et al. in [16]. Nevertheless, these strategies are primarily applicable for private cloud setups, hence these strategies are criticised by a larger research community. The moderated strategy parallel to this is proposed by C. C. Tarek Menouer et al. [18] to overcome the challenges of isolated cloud. Further the work of C.C. Tarek Menouer et al. [18] is enhanced by Keqin Li et al. [10] by using the Multiple Heterogeneous Servers specialized for edge computing.

The working models of these proposed strategies are numerous as one of the most popular deployment is described by Yue Xu et al. in [20] for traffic management and prediction based on the proposed framework by Longjiang Li et al. in [11].

Nonetheless, the demands and purposes for the IoT networks have came a long way from the initial research outcomes. The present IoT networks are utilized for higher and higher capacitive situations. The data collected by the network nodes have also grown in volume and the applications handling the data must also be scaled. Hence, the practitioners have started utilizing the benefits from cloud computing. The research outcomes from the cloud computing have proposed a good number of solutions to balance the load for cloud specific applications handling the data from the cloud services. Nevertheless, the algorithms are not designed and not suitable for handling applications dealing with IoT data.

Henceforth, this work proposes a novel framework for balancing loads for cloud integrated IoT framework. The novel algorithm is elaborated in the next section of this work.

5 IoT cloud integrated load balancing algorithm

The proposed framework deploys an algorithm for migrating the highly loaded virtual machine to a lesser loaded destination physical host. The proposed algorithm utilizes 4 major properties of virtual machines handling IoT node data as CPU utilization, Power consumption or the energy, memory utilization and finally the SLA violation due to the overload.

Cloud stack adjusting is the way toward disseminating remaining tasks at hand and registering assets in a cloud processing condition. Load adjusting enables ventures to oversee application or remaining task at hand requests by dispensing assets among various PCs, systems or servers. Cloud stack adjusting includes facilitating the circulation of outstanding burden traffic and requests that live over the Internet. The algorithm is furnished here.

From the understanding of the parallel research outcomes, it is natural to realise that most of the load balancing algorithms have failed in justifying the IoT network load balancing demands as all the parameters involved in analysing the load of any virtual machine were given equal weightage. In the contradiction, this work proposes a novel weightage metric to justify the IoT load balancing demands as studied in the previous sections of this work.

Considering the limited power and processing capabilities of any IoT node in the network, this algorithm improves the energy efficiency and the processing capabilities to a greater extend. Also, the loading balancing process time complexity is reduced significantly.

The working capabilities of the proposed ESBL algorithm with existing IoT framework is elaborated in the next section of this work.

6 Proposed framework for cloud integrated IOT load balancing

In this section of work, the proposed framework to balance load for cloud integrated IoT network is elaborated. The proposed ESBL algorithm can get embedded in the IoT function

Algorithm 1 Energy sensitive for balancing load algorithm (ESBL)

```
Initialize the IoT network routing table
Initialize all nodes
Accumulate the data from the IoT nodes into Cloud
Instantiate the Virtual machines
Analyse the present load as VM[i]
For each virtual machine
a. Calculate CPU utilization as CPUV[i]
b. Calculate power consumption as PowerV[i]
c. Calculate Memory utilization as MemV[i]
d. Calculate SLA violation as SlaV[i]
For each VM[i]
a. calculate the LoadV[i] = (CpuV[i] * 0.4) + (PowerV[i] * 0.1) +(MemV[i] * 0.3) + (SlaV[i] * 0.2)
For LoadV
a. IF $LoadV[i] > LoadV[j]$
b. $High_V M \leftarrow LoadV[i]$
c. $Low_V M \leftarrow LoadV[j]$
$SourceVM \leftarrow High_V M$
$DestinationVM \leftarrow Low_V M$
$CalculateVM_S hutDown$
$CalculateEnergyV \leftarrow PowerV[DestinationVM]$
$MigrateVM[DestinationVM]$
```

process. The proposed modified framework is elaborated in Figure 3.

In the initial phase of the network operations, the network components must be started and generate the beacon signal as part of the initial bootstrap process. The IoT involves extending Internet connectivity beyond standard devices, such as desktops, laptops, smart phones and tablets, to any range of traditionally dumb or non-internet-enabled physical devices and everyday objects. Embedded with technology, these devices can communicate and interact over the Internet, and they can be remotely monitored and controlled. All the devices associated with the network must be up and running in this initial phase of the framework and the framework is also responsible for assuring connectivity between the components.

In the next phase of the framework, the network routing table must be updated to ensure the appropriate data migration to the cloud. A routing table uses a similar thought that one does when utilizing a guide in bundle conveyance. At whatever point a hub needs to send information to another hub on a system, it should initially realize where to send it. If the hub can't straightforwardly associate with the goal hub, it needs to send it through different hubs along an appropriate course to the goal hub. Most hubs don't Endeavor to make sense of which course may work; rather, a hub will send an IP bundle to an entryway in the LAN, which at that point chooses how to course the "bundle" of information to the right goal. Every door should monitor which approach to convey different bundles of information, and for this, it utilizes a routing table. A routing table is a database which monitors ways, like a guide, and uses these to figure out which approach to forward traffic. Entry ways can likewise share the substance of their routing table with different hubs asking for that data.

Furthermore, the network controller using this framework must migrate all the data collected by the network to be migrated to the cloud hosts. The applications running on the cloud hosts shall use these data for further purposes.

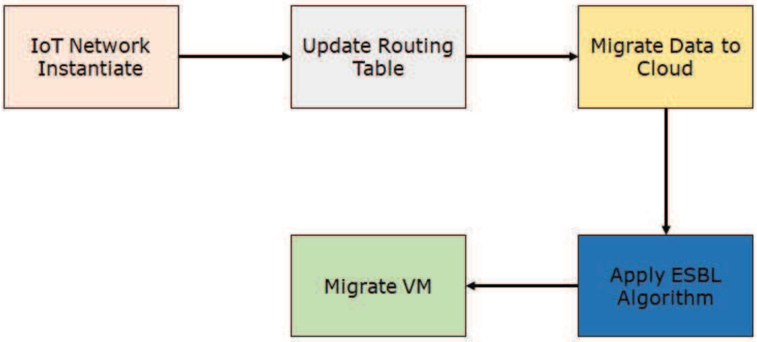

Figure 3: Proposed cloud integrated IoT application load balancing process flow

Table 1: Initial experimental setup

Algorithm Name	No. of hosts	Number of VMs	Total simulation time(sec)
IQR MMT	50	100	86400
IQR MU	50	100	86400
LR MMT	50	100	86400
LR MU	50	100	86400
MAD MMT	50	100	86400
MAD MU	50	100	86400
THR MMT	50	100	86400
THR MU	50	100	86400
ESLB	50	100	86400

Once the applications are up and running, the ESBL algorithm, will identify the heavily loaded source VM host instance and less loaded destination physical host. Finally, after the identification of source and destination nodes, the migration happens.

The framework continues to collect data from the IoT network and performs the load balancing strategies as and when imbalanced nodes are identified.

7 Results and discussions

The results obtained from the proposed algorithm and the cloud integrated framework are highly satisfactory. The obtained results are furnished in this section of the work and discussed further.

The results obtained from the proposed framework is analysed in few different parts as load analysis of the IoT network before and after the migration, Initial virtual machine setups, Energy consumption, Number of node shutdowns and finally Execution time of the total load balancing time. During the analysis of the results, the results obtained from the proposed algorithm is also compared with other parallel research algorithms for cloud-based load balancing.

7.1 Initial virtual machine setups

Firstly, the initial virtual machine setups are discussed in this section. All the algorithms, proposed and comparable, are tested under a similar situation. The initial setup is discussed in Table 1.

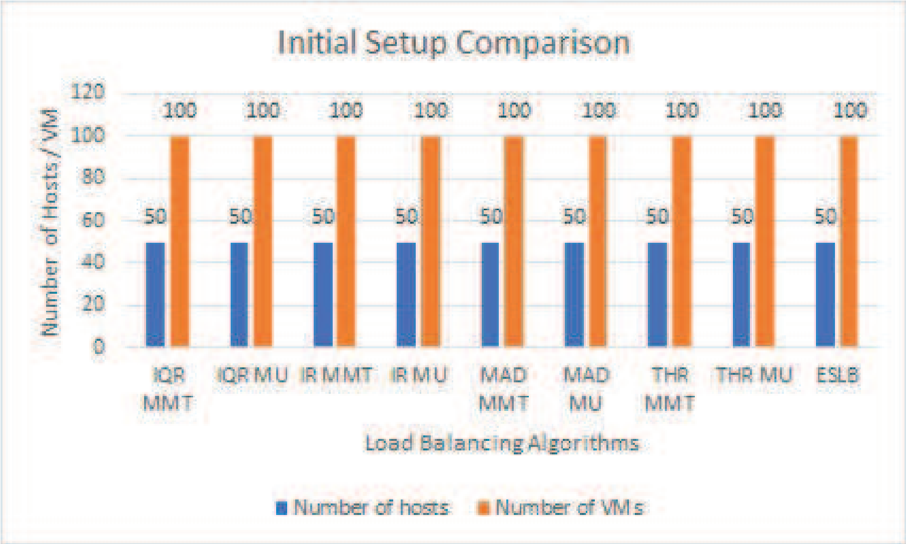

Figure 4: Initial setup comparisons

Table 2: Energy consumption analysis

Algorithm Name	Energy Consumption (KWh
IQR MMT	47.85
IQR MU	49.32
LR MMT	35.37
LR MU	35.38
MAD MMT	45.61
MAD MU	47.36
THR MMT	41.81
THR MU	44.08
ESLB	36.81

The detailed of the comparative algorithms are discussed in the further section of this work. The results are also visualized graphically in Figure 4.

7.2 Energy consumption

Secondly, the energy consumption by the virtual machines depend on the energy consumption for running the applications on the virtual machines and the load balancing. The calculation of the energy or power consumption is calculated for the complete virtual structure of the network, where all the virtual machines are included. This strategy helps in realizing the effective utilization of the power.

The load balancing algorithms are intended to reduce the load on one single physical instance or a specific virtual machine to reduce the power consumption. The power consumption of the proposed algorithm is analysed and compared with the other parallel standard load balancing algorithms [Table 2]. The results ate visualized graphically in Figure 5.

It is natural to understand that the overall energy consumption of the virtual network for IoT application and data management is significantly less compared to the other standard applications.

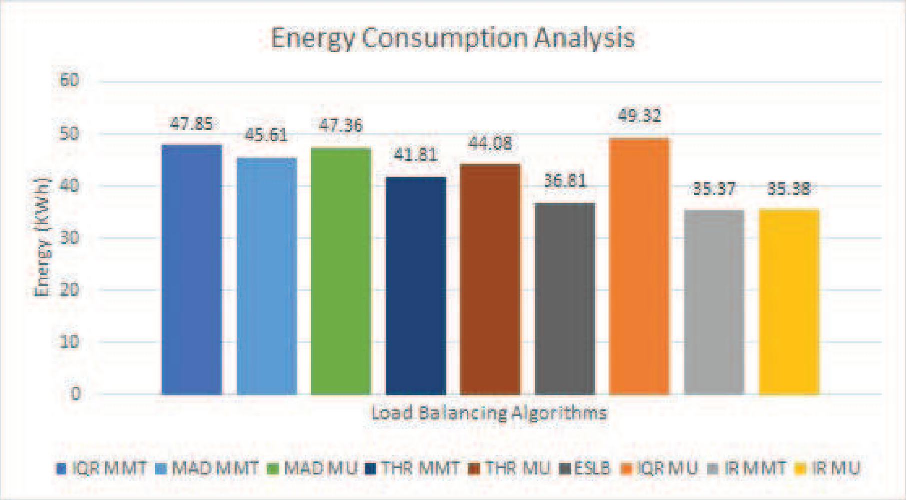

Figure 5: Energy consumption analysis

Table 3: Number of node (physical hosts) shutdowns

Algorithm Name	Number of node shutdowns
IQR MMT	1549
IQR MU	1622
LR MMT	806
LR MU	816
MAD MMT	1528
MAD MU	1632
THR MMT	1424
THR MU	1578
ESLB	795

7.3 Number of physical hosts shutdowns

Thirdly, the number of physical hosts during the idle phase tend to shutdown if for a longer duration no load is been assigned. Hence, the poorest load balancing algorithm will have maximum number of host shutdowns. The Physical hosts shutdown analysis is furnished in Table 3.

The results are been analysed graphically in Figure 6.

It is natural to realize that, the proposed algorithm is demonstrating much lesser hosts shutdowns, which clearly signifies that the utilization of the available resources is significantly high compared with the other algorithms.

7.4 Load balancing execution time

Fourthly, the load balancing algorithm time complexity is analysed. The code responsible for balancing the loads are also deployed on the same physical host instances and buy out the time required to run, or time required for application to respond. The time complexity is analysed in Table 4.

The results are been analyzed graphically in Figure 7.

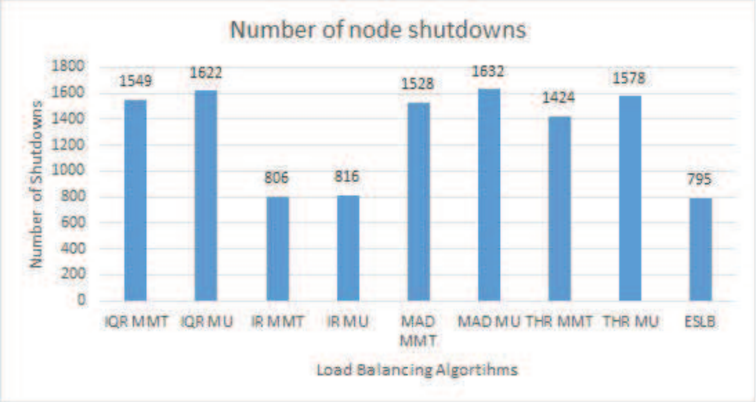

Figure 6: Host N node shutdown analysis

Table 4: Load balancing execution time

Algorithm Name	Execution time (Sec)
IQR MMT	0.002510
IQR MU	0.003100
LR MMT	0.001680
LR MU	0.001530
MAD MMT	0.003310
MAD MU	0.014710
THR MMT	0.001330
THR MU	0.001600
ESLB	0.001560

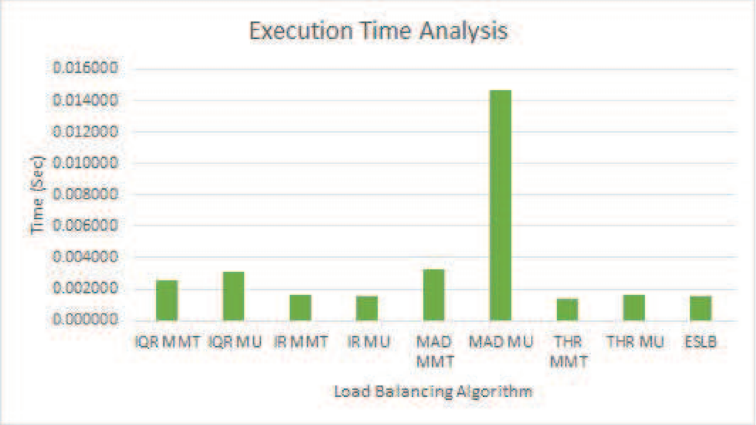

Figure 7: Execution time analysis

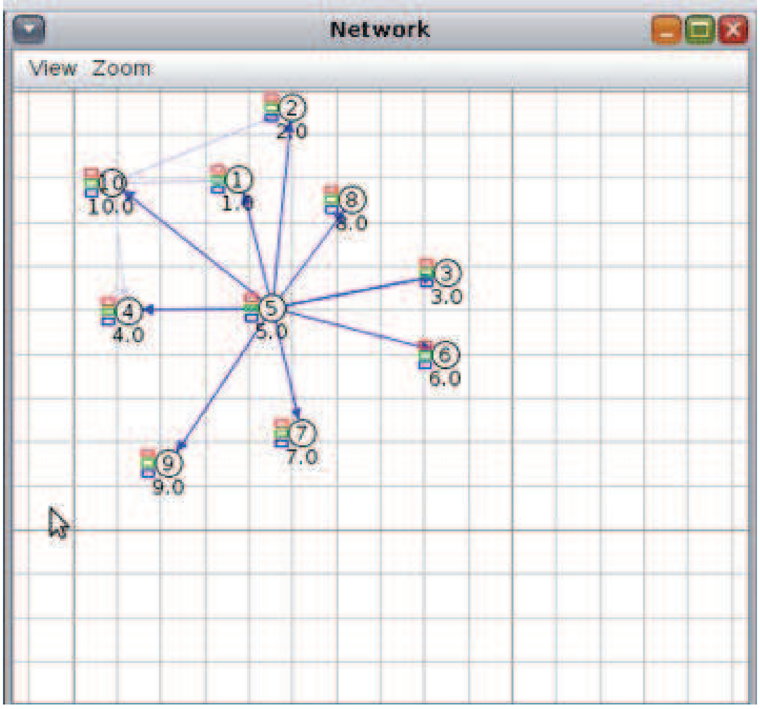

Figure 8: Initial insanitation of the network

7.5 IoT network performance

Finally, the performance of any IoT network can be measured by analysing the responsiveness of the network. During a network performance improvement process, if the response time or the number of response requests in the same time increases, then it is natural to accept that the performance of the network has improved.

The network is observed during the routing phases and some of the scenarios are furnished in Figures 8 – 10.

The performance comparison of the configured IoT network is analysed in Table 5.

Table 5: Load balancing effects on IoT network

Source Node	Number of Connect Requests	Average Response with ESBL	Average Response without ESBL
1	491	2.063	1.941
2	390	2.086	1.921
3	297	2.092	1.916
4	398	2.187	1.843
5	971	2.093	1.915
6	393	2.015	1.985
7	291	2.109	1.902
8	491	2.098	1.911
9	285	2.192	1.839
10	395	2.147	1.872

The improvement is visualized graphically in Figure 11.

It is natural to observe that the response time have significantly improved without compro-

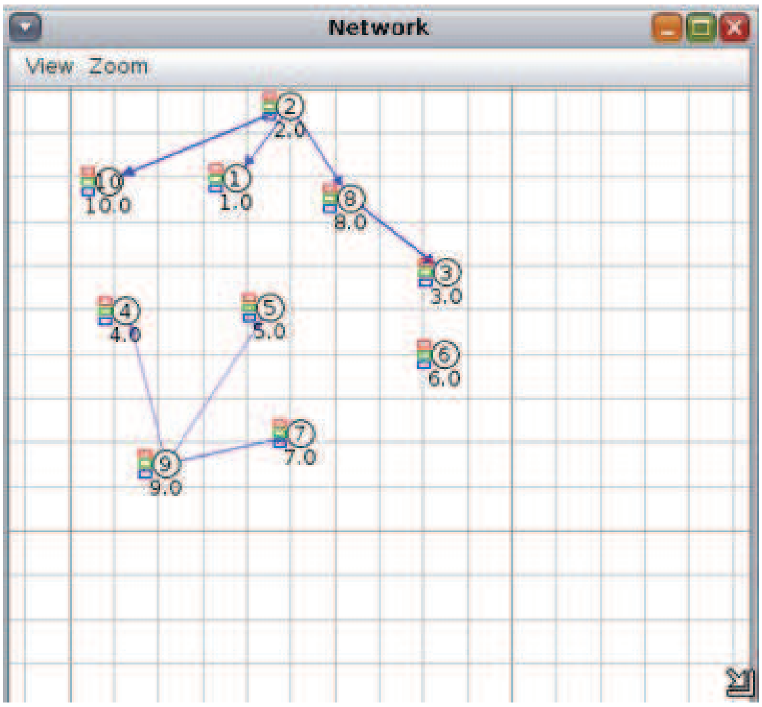

Figure 9: Routing at time instance T1

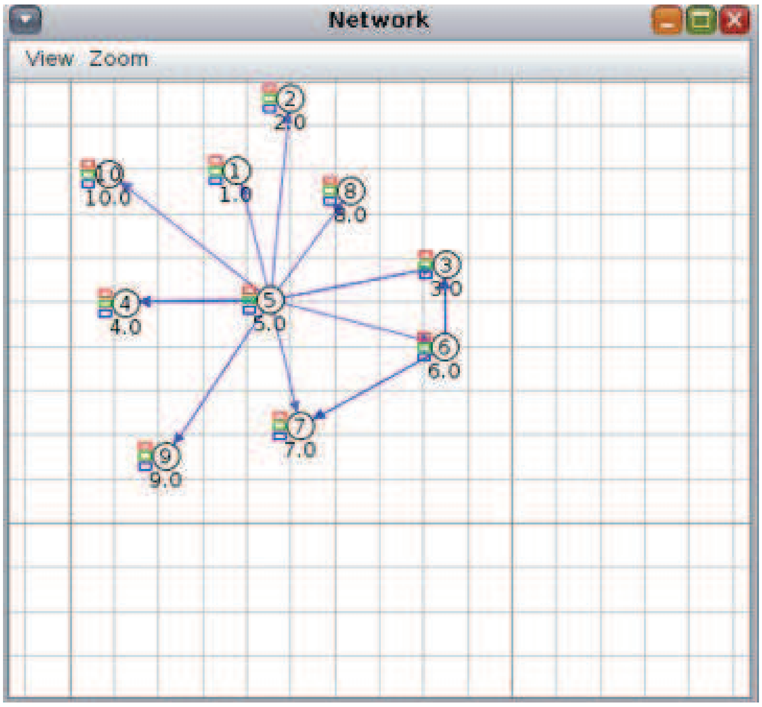

Figure 10: Routing at time instance T2

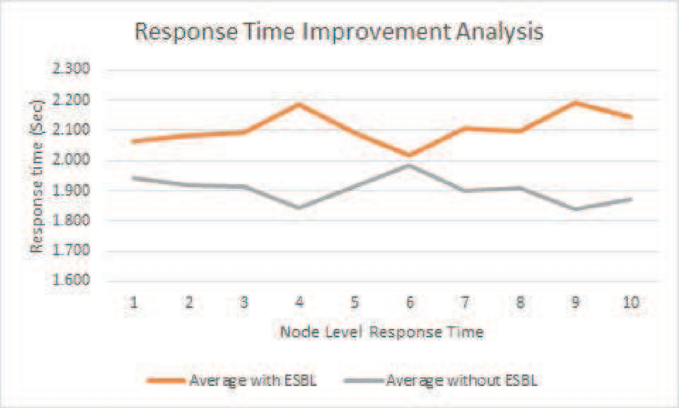

Figure 11: Response time improvement analysis

mising the processing capabilities and energy efficiently.

In the next section of the work, the comparative analysis is carried out.

8 Comparative analysis

To realize the benefits or improvements obtained from any proposed algorithm or framework can be understood by comparing the results with parallel or standard research outcomes. Hence in this section of the work, the comparative analysis is carried out [Tab. 6].

From the comprehension of the parallel research results, it is normal to understand that the vast majority of the heap adjusting calculations have flopped in advocating the IoT organize load adjusting requests as every one of the parameters associated with examining the heap of any virtual machine were given equivalent weight age. In the logical inconsistency, this work proposes a novel weight age metric to legitimize the IoT load adjusting requests as concentrated in the past areas of this work. Considering the constrained power and handling capacities of any IoT hub in the system, this calculation improves the vitality effectiveness and the preparing abilities to a more noteworthy expand. Likewise, the stacking adjusting process time multifaceted nature is decreased altogether.

Table 6: Comparative analysis

Name	Selection Policy	Allocation Policy	Comparison ESLB		
			Energy consumption	Number of node shutdowns	Execution Time
IQR MMT	Minimum Migration Time	Inter Quartile Range	Improved	Improved	Improved
IQR MU	Minimum Utilization	Inter Quartile Range	Improved	Improved	Improved
LR MMT	Minimum Migration Time	Local Regression	Not Improved	Improved	Improved
LR MU	Minimum Utilization	Local Regression	Not Improved	Improved	Not Improved
MAD MMT	Minimum Migration Time	Median Absolute Deviation	Improved	Improved	Improved
MAD MU	Minimum Utilization	Median Absolute Deviation	Improved	Improved	Improved
THR MMT	Minimum Migration Time	Static Threshold	Improved	Improved	Not Improved
THR MU	Minimum Utilization	Static Threshold	Improved	Improved	Improved

Hence, it is natural to realize that the proposed algorithm has outperformed most of the parallel research outcomes.

9 Conclusions

The increasing demand for higher complex data processing applications for managing IoT network data make motivated the application and network designer to push the application on to the cloud computing environment. Considering less dependencies of services and platform from the cloud service models, the major need is consisting of only infrastructure from the cloud service type offerings. With this decision of utilizing the benefits of infra as a service, the demand for balancing the loads on cloud computing is also unavoidable. Observing the parallel standard load balancing strategies, this work identifies that the existing algorithms are not designed to match the demands of IoT networks. Hence this work proposes a novel algorithm for balancing loads for cloud integrated IoT frameworks utilizing the characteristics of loads such as CPU utilization, memory utilization, energy consumption, SLA non-violation and equal distribution of the virtual machines with equal distributions of application loads on virtual machines. The algorithm demonstrates significant improvements over standard algorithms and also improves the IoT network response time by 60.

Bibliography

[1] Ahanger, T.A. (2018). Defense Scheme to Protect IoT from Cyber Attacks using AI Principles, *International Journal of Computers Communications & Control*, 13(6), 915–926, 2018.

[2] Alvi, S. A.; Shah, G. A.; Mahmood, W. (2015). Energy efficient green routing protocol for internet of multimedia things, *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, IEEE, 1–6, 2015.

[3] Balakrishna, G.; Rao, M. N. (2019). Study report on using iot agriculture farm monitoring. *Innovations in Computer Science and Engineering*, Springer, 483–491, 2019.

[4] Di Marco, P.; Athanasiou, G.; Mekikis, P.-V.; Fischione, C. (2016). Mac-aware routing metrics for the internet of things, *Computer Communications* 74, 77–86, 2016.

[5] Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything, *CISCO white paper* 1–11, 2011.

[6] Guo, J.; Orlik, P.; Zhang, J.; Ishibashi, K. (2014). Reliable routing in large scale wireless sensor networks, *14 Sixth International Conference on Ubiquitous and Future Networks*,IEEE, 99–104, 2014.

[7] Karthikeya, S. A.; Vijeth, J.; Murthy, C.S.R. (2016). Leveraging solution-specifc gateways for cost-effective and fault-tolerant iot networking, *IEEE Wireless Communications and Networking Conference*, 1–6, 2016.

[8] Kim, H.-Y. (2015). An effective load balancing scheme maximizes the lifetime in wireless sensor networks, *5th International Conference on IT Convergence and Security (IC-ITCS)* 1–3, 2015.

[9] Le, Q.; Ngo-Quynh, T.; Magedanz, T. (2014). Rpl-based multipath routing protocols for internet of things on wireless sensor networks, *International Conference on Advanced Technologies for Communications (ATC 2014)*, 424–429, 2014.

[10] Li, K. (2019). Computation Offloading Strategy Optimization with Multiple Heterogeneous Servers in Mobile Edge Computing, *IEEE Transactions on Sustainable Computing*, 1-1, 2019.

[11] Li, L.; Zhou, H.; Xiong, S. X.; Yang, J.; Mao, Y. (2019). Compound model of task arrivals and load-aware offloading for vehicular mobile edge computing networks, *IEEE Access*, 7, 26631–26640, 2019.

[12] Mainetti, L.; Patrono, L.; Vilei, A. (2011). Evolution of wireless sensor networks towards the internet of things: A survey, *SoftCOM 2011, 19th international conference on software, telecommunications and computer networks*,IEEE, 1–6, 2011.

[13] Menouer, T.; Cerin, C. (2017). Scheduling and resource management allocation system combined with an economic model, *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications*, 807–813, 2017.

[14] Oteafy, S. M.; Al-Turjman, F. M.; Hassanein, H. S. (2012). Pruned adaptive routing in the heterogeneous internet of things, *In 2012 IEEE Global Communications Conference(GLOBECOM)*, IEEE, 214–219, 2012.

[15] Oteafy, S. M. and Hassanein, H. S. Towards a global iot: Resource re-utilization in wsns. *In 2012 international conference on computing, networking and communications (ICNC), IEEE* pages 617–622.

[16] Peinl, R.; Holzschuher, F.; Pfitzer, F. (2016). Docker cluster management for the cloud-survey results and own solution, *Journal of Grid Computing*, 265–282, 2016.

[17] Petrioli, C.; Nati, M.; Casari, P. et al. (2013). Alba-r: Load-balancing geographic routing around connectivity holes in wireless sensor networks, *IEEE Transactions on Parallel and Distributed Systems*, 25(3), 529–539, 2013.

[18] Tarek Menouer, C.C.; Leclercq, E. (2018). New multi-objectives scheduling strategies in docker swarmkit. *In International Conference on Algorithms and Architectures for Parallel Processing*, Springer, 103–117, 2018.

[19] Wu, M.; Lu, T.-J.; Ling, F.-Y. et al. (2010). Research on the architecture of internet of things, *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, IEEE, 5, 484–487, 2010.

[20] Xu, Y.; Yin, F.; Xu, W. et al. (2019). Wireless Traffic Prediction with Scalable Gaussian Process: Framework, Algorithms, and Verification, *IEEE Journal on Selected Areas in Communications*, 37(6), 1291–1306, 2019.