IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

ENERGY EFFICIENT USING MILP FOR PEERS VIRTUALIZATION FRAMEWORK

[1]K TEJASHIWINI, [2] Mr G.VENKATA RAMI REDDY, [3] Mr KATROTH BALAKRISHNA

[1]Student, [2] Professor, [3] Assistant professor

[1]Software Engineering,

[1]School of Information Technology, Hyderabad, India

Abstract: An energy efficient IoT virtualization framework with peer-to-peer (P2P) networking and edge processing is proposed. In this network, the IoT task processing requests are served by peers. IoT objects and relays that host virtual machines (VMs) represents the peers in the proposed P2P network. We have considered three scenarios to investigate the saving in power consumption and the system capabilities in terms of task processing. The first scenario is a 'relays only' scenario, where the task requests are processed using relays only. The second scenario is an 'objects only' scenario, where the task requests are processed using the IoT objects only. The last scenario is a hybrid scenario, where the task requests are processed using both IoT objects and VMs. We have developed a mixed integer linear programming (MILP) model to maximize the number of processing tasks served by the system, and minimize the total power consumed by the IoT network. Based on the MILP model principles, we developed an energy efficient virtualized IoT P2P networks heuristic (EEVIPN). Our results show that the hybrid scenario serves up to 77% (57% on average) processing task requests, but with higher energy consumption compared to the other scenarios. The relays only scenario serves 74% (57% on average) of the processing task requests with 8% saving in power consumption compared to the hybrid scenario. In contrast, 28% (22% on average) of task requests can be handled by the objects only scenario with up to 62% power saving compared to the hybrid scenario.

***Index Terms* – Power saving,processing task request, Scenarios.**

I. INTRODUCTION

The various recently developed IoT are driven by the tremendous needs and benefits are made by connecting physical world to the Internet. It is expected that there can be 50 billion (and by some estimates, more) or more devices in the coming years [1]. This growth improves the number in connected devices and give chance to new applications, the example can be manufacturing, smart homes, transportation, agriculture, and M2M communications [2], [3].There are many challenges like energy efficient, reliable, secure, interoperable and scalable. So, we have to realize and overcome this before the planned growth in the number and functionalities [4]. From the expected number of devices, one of the most significant challenges is energy efficiency by greening the associated networks, that grabbed attention for the academic and industrial domains .To provide the solution Cloud computing helps in challenge for network and data centers [5][8]. The work started with distributing the content placement, by bringing content closer to users [10] and data centers, thus bringing the process of capabilities [11] and distributing the process of big data, where processing of generating the huge data by IoT devices nearer source will extract knowledge and transmit the small volume as `extracted knowledge' message, thus saving the network and process the resources and energy [12]. Thus, an efficient solution is advocated here this will be the IOT data that processed by IOT objects or by the nearer devices layer to objects.

1.1 EXISTING SYSTEM

In existing frameworks, perhaps the most test is vitality effectiveness and thus greening the related systems, which caught eye in both the scholarly and modern spaces. Distributed computing is researched as one of the answers for the vitality effectiveness challenge in systems and server farms with the high information created by the associated IoT objects, rising distributed computing with IoT presents new difficulties which must be tended to. Among these difficulties is the long for we need all the more preparing abilities, high correspondence data transmission, security, and inertness necessities.

1.1.1 DISADVANTAGES OF EXISTING SYSTEM

1. Security is less.
2. Communication failures are occur.

1.2 PROPOSED SYSTEM

In this paper, a vitality productive IoT virtualization system with shared (P2P) systems administration and edge handling is proposed. In this system, the IoT task preparing demands are served by peers. IoT items and transfers that have virtual machines (VMs) speaks to the companions in the proposed P2Pnetwork.We have considered three situations to research the sparing in power utilization and the system capabilities in terms of task processing. The first situation is a relays only scenario, where the tasks demands are handled utilizing relays only. The subsequent situation is an objects only, where the undertaking demands are prepared utilizing the IoT objects only. The last situation is a hybrid situation, where the undertaking demands are handled utilizing both VMs and IoT objects. We have built up a MILP model to expand the quantity of handling tasks served by the framework, and limit the total power consumption by the IoT network. Based on MILP model standards, we built up a energy effective virtualized IoT P2P systems heuristic (EEVIPN).

1.2.1 ADVANTAGES OF PROPOSED SYSTEM

- Saving the power consumption.

II. SYSTEM REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

- Processor : Core – i4
- RAM : 4GB
- Hard Disk :1TB

2.2 SOFTWARE REQUIREMENTS

- Coding Language : Java
- IDE : Eclipse
- Operating System : Windows 10

III. RELATED WORK

Practically evaluating Internet of Things by abstraction technique and information processing [1]

The word IoT means to communication and interaction between producing of data relate to real world objects (like things) and billions devices that exchange. The devices which captures the data extract of high level data from the raw sensor and represents this data as human-understandable or machine-interpretable has many interesting applications. Deriving higher level information by raw data represents demand to extract, characterize and find, the raw data into meaningful abstractions. These meaningful abstractions which are present in a machine-understandable and/or human representation.

Future edge computing and edge cloud for IOT applications [2]

Towards the future of IOT the evolving of Internet rapidly make potential connects of billions or even more of edge devices that can make high amount of information at a very high speed and few of the applications may require very low latency. The traditional infrastructure cloud run into set of problems due to networking, storage and centralized computation in a small set of datacenters, the remote datacenters and the edge devices due to long distance between. To face this challenge, edge computing and edge cloud have to be possibility that gives resources even nearer to the poor edge IOT resource and a new IoT innovation ecosystem potentially to nurture. Such prospect by a set of emerging technologies is enabled, includes networking defined software and network function virtualization. Here, key rational, research topics, the key enabling technology and the efforts of art-of-the-state are investigated and typical IoT applications are benefited from edge cloud. We see to aim whole picture of ongoing and next comprehensive discussions of possible research directions.

Internet of Things : A starting applications, protocols and technologies on survey [3]

This project gives an whole view on the IoT which emphasis on application issues, enabling technologies and protocols. The IoT enables the RFID development. The premise is to send a new applications collaborate directly having smart sensors without human involvement. The present mobile, revolution in Internet and M2M (machine-to-machine) technology seen as the 1st stage of the IoT.

IV. PRODUCT DESIGN - UML DIAGRAMS:

4.1 UseCase Diagram

Usecase diagram is a UML diagram that indicates the relationships between users, actors and systems to catch the requirements of system, validate system architecture, specifies the context of a system. Use case diagram consists of actors, use-cases, communication link and the boundary of the system. Actor is the one who interacts with the system or usecases , here actor is the user(ourselves) and usecase generally represents the functionality of the system(process- automated or manual), here usecases are upload facebook data, preprocess and clustering, intimacy matrix, information flow miner,response weight graph.Communication link is the solid line connecting an actor to a usecase.

Fig -1: UseCase diagram

4.2 Sequence Diagram

Sequence diagram is a UML diagram that indicates the flow of actions among users here, among user and the system. The user interacts with the system as shown below.First, he/she uploads data consisting of Facebook post, after posts data are successfully loaded, upload Facebook comments data, next comments and posts is pre-processed and clustered into groups by removing noisy users. Then, intimacy matrix is calculated to know the closeness between users. The matrix displays the data of users commenting on posts, then apply information flow miner to get the information between two users and their weights. Lastly, a graph is generated.

Fig -2: Sequence diagram

4.3 Activity Diagram

Activity diagram is a UML diagram which is essentially an improved form of flow chart representing flow from one action or activity state to another state .Unlike flowchart ,it consists of initial state which is a black filled circle as shown below. Action or activity states which is represented as rectangle with rounded corners, and control flow referred to as paths which represents transition from one action or activity state to another state. Decision node and branching used to make decision before determining the flow of control. The activity diagram for information flow miner is as shown below:

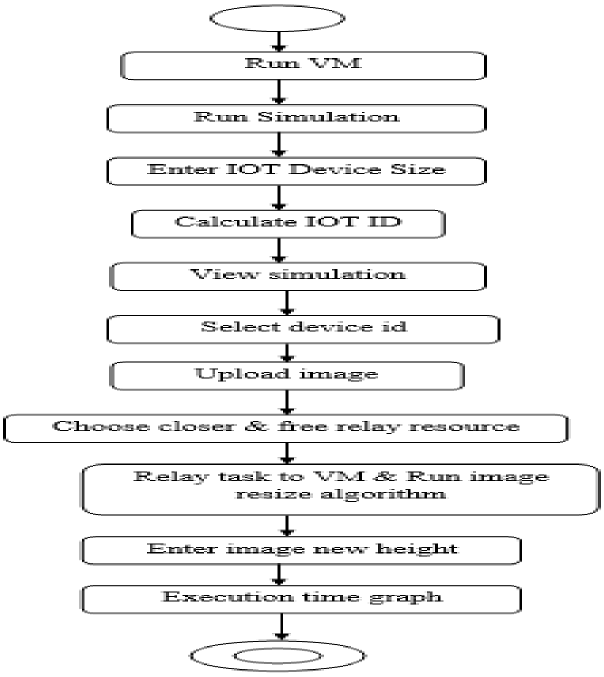


Fig -3: Activity diagram

4.4 DataFlow Diagram

Dataflow diagrams is a graphical form of logical flow of information in the system I.e; between a system and components of a system. It consists of process which receives input and produces output and generally represented in the form of rounded rectangle and, the other one is data-flow which is a path for data to move from one component to other and represented as straight lines with either incoming or outgoing arrows. The data flow diagram for information flow is as follows: user and system are two processes and arrows represent the data-flow between them.

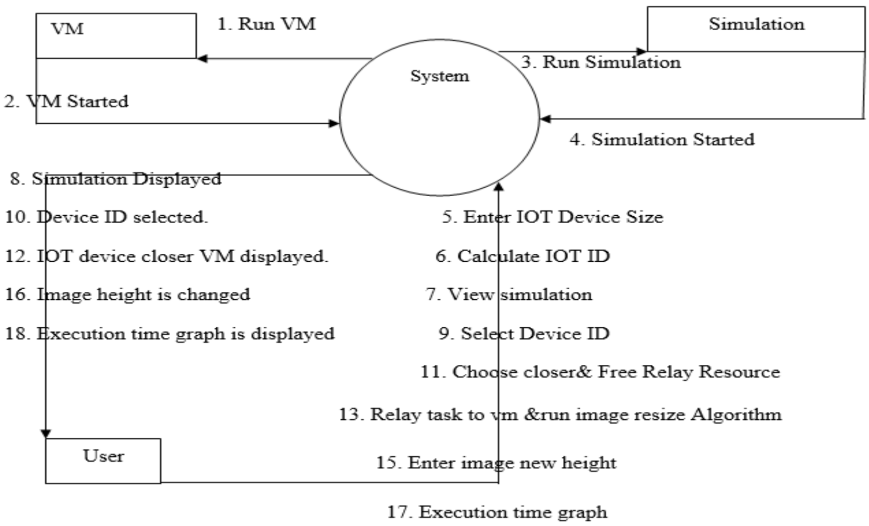
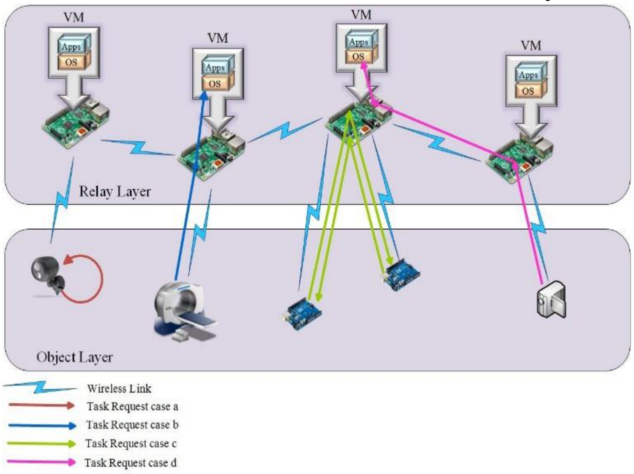


Fig -4: Dataflow diagram

V. Implementation

5.1 ENERGY EFFICIENT MILP FOR P2P IoT NETWORKS

The MILP model developed considers the architecture shown in Fig. 1. The proposed architecture is constructed of two lay- ers. The first layer represents the IoT objects. The upper layer consists of the relay devices that realize traffic transportation between peers. In our framework, each object is capable of processing three types of tasks that are required by other objects. The task processing capabilities and task require- ments for the IoT objects are specified by the MILP model parameters. Each relay node has the ability to host VMs in order to process the tasks requested by IoT objects. The number of relays that can handle all task types is limited to a subset of total number of relays. For example, in the results section we consider a scenario in which 10 out of 25 relays host VMs that can handle all tasks types.



5.2 Algorithm Description

In this section, an energy efficient virtualized IoT P2P networks (EEVIPN) heuristic algorithm has been developed for real time implementation and to verify the MILP model results. The EEVIPN heuristic is illustrated in Fig. 8. It con- siders the hybrid scenario as it is the generic scenario that can be used to build other scenarios such as the relays only and the object only scenarios. To determine the total power consumption (TPC), the heuristic determines the type and the optimum place of the peer to be used to serve the processing tasks according to the serving constraints of each peer. The serving constraints can be summarized as follows:

i. The processing task should not have been served by any other peer before.
ii. The upload traffic of each candidate peer should not exceed the maximum limit.
iii. The download traffic of each candidate peer should not exceed the maximum limit.
iv. The upload slots of each object should not exceed the specified maximum number
v. The number of candidate relays hosting VMs should not exceed the specified maximum number of serving relays.
vi. There should be sufficient processor capacity in each candidate peer to accommodate the processing task workload.

Inputs:
$O = \{1 \dots NO\}$
$K = \{1 \dots NK\}$
$R = \{1 \dots NR\}$
Output: No. of Served Tasks
Total Power Consumption (TPC)
For each task k $\in K$ Do
For each object requesting a task $i \in O$ Do
For each candidate relay hosting VM that can serve a requested task $j \in R$ Do
If all serving constraints are met Then
U(i,j,k)=1
Calculate P_j^{rp}
End If
End For
End For
End For
For each task k $\in K$ Do
For each object requesting a task $i \in O$ Do
For each candidate object that can serve a task $j \in O$ Do
Case $(i = j)$
If all serving constraints are met Then
U(i,j,k)=1
Calculate P_j^{op}
End If
End Case
Case $(i \neq j)$
If all serving constraints are met Then
Do Tit for Tat
U(i,j,k)=1
Calculate P_j^{op}
End If
End Case
End For
End For
End For
For Each IoT object $i \in O$ Do
Calculate P_i^{otr}
End For
For Each relay sending and receiving traffic to and from other relays $(a \in R)$
Calculate P_a^{rtr} based on minimum hop path between node pair (x,y)
End For
For each relay receiving task requests from objects and sending task results to objects $(a \in R)$
Calculate P_a^{rtr}

V. Testing and Results

The procedure or strategy for discovering errors or defects in an application or software program with the goal that the application functions as indicated by the end client's prerequisite is called testing. The test case is defined as a set of conditions or factors under which a tester will decide if a system or application under test satisfies requirements and works properly.

Following are the testing strategies followed during the test period:

TEST CASES:

Test Case Id	Test Case Name	Test Case Desc.	Test Steps			Test Case Status	Test Priority
			Step	Expected	Actual		
01	Run VM and Simulator	Verify the VM and Simulator started or not	Without VM and Simulator	Users cannot do further operations	VM and Simulator are started	High	High
02	Enter IOT Device size	Verify IOT Device size is enter or not	Without entering the IOT Device size	It cannot display the IOT Device size	It can display the IOT Device size	High	High
03	View Simulation	Verify the simulation is displayed or not	Without assigning each id to device	The simulation cannot displayed	The simulation can displayed	High	High
04	Upload Image	Verify the image is uploaded or not	Without selecting any device id	we cannot upload image	selected image will send to vm	High	High
05	Choose closer &Free Relay Resource	Verify closer &Free Relay Resource are finded or not	without run this Choose closer &Free Relay Resource	we cannot find free and closer relay/vm	we cannot find free and closer relay/vm	High	High
06	Relay Task o vm & Run Image Resize Image Resize Algorithm	Verify the node behavior chart is displayed or not	Without saving the abnormal weight of the sensors	The Node behavior Chart is not displayed	The Node behavior Chart is displayed successfully	High	High
07	Relay Task to vm & Run Image Resize Image Resize Algorithm	Verify Relay Task to vm & Run Image Resize Image Resize Algorithm run or not	Without finding closer relay/vm	The algorithm will not run	The algorithm will run and allow to enter image new height	High	High

08	Execution Time Graph	Verify the node behavior chart is displayed or not	Without saving the abnormal weight of the sensors	The Node behavior Chart is not displayed	The Node behavior Chart is displayed successfully	High	High

Table -1: Test cases

RESULTS:

First double click on 'run.bat' file from 'VM' folder to get below screen and let it run

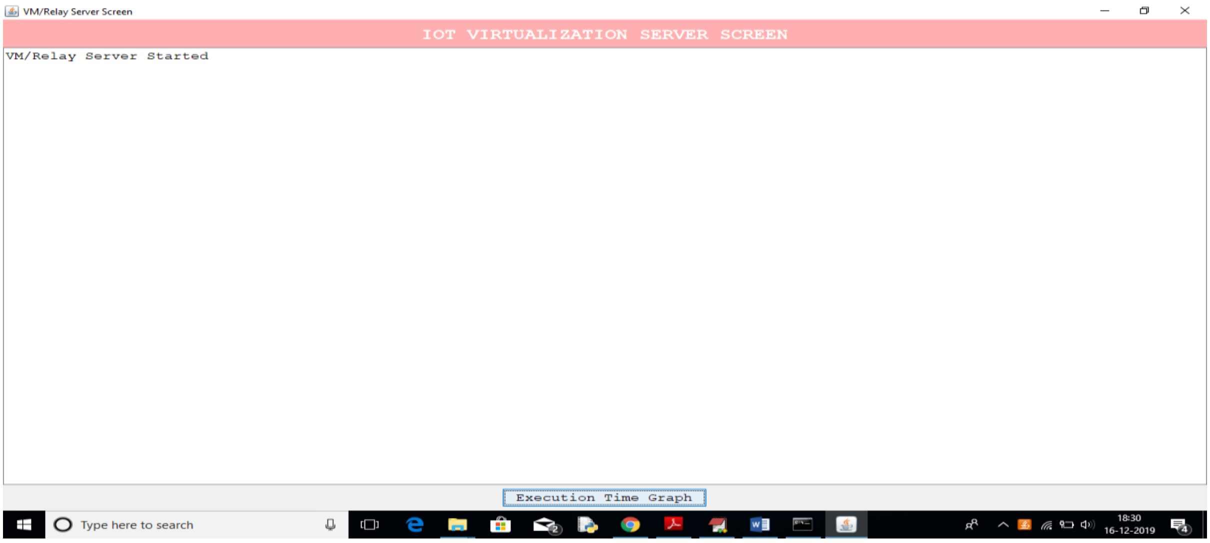


Fig: VM start server

Now double click on 'run.bat' file from 'Simulation' folder to get below screen

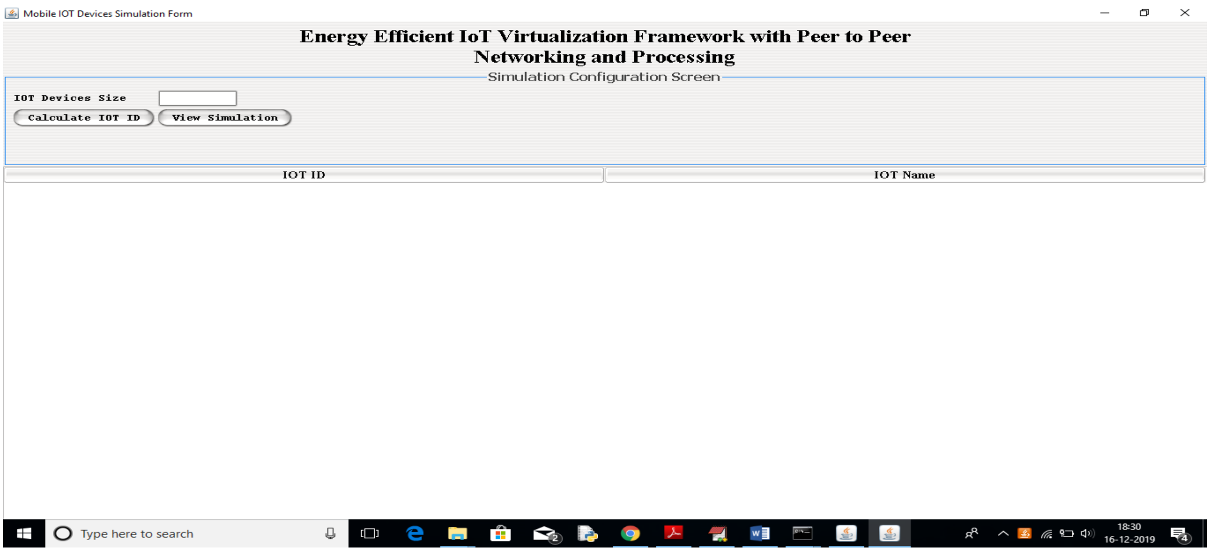


Fig: Starting of simulation

In above screen enter IOT Devices Size and then click on 'Calculate IOT ID' button to assign ID to each IOT device

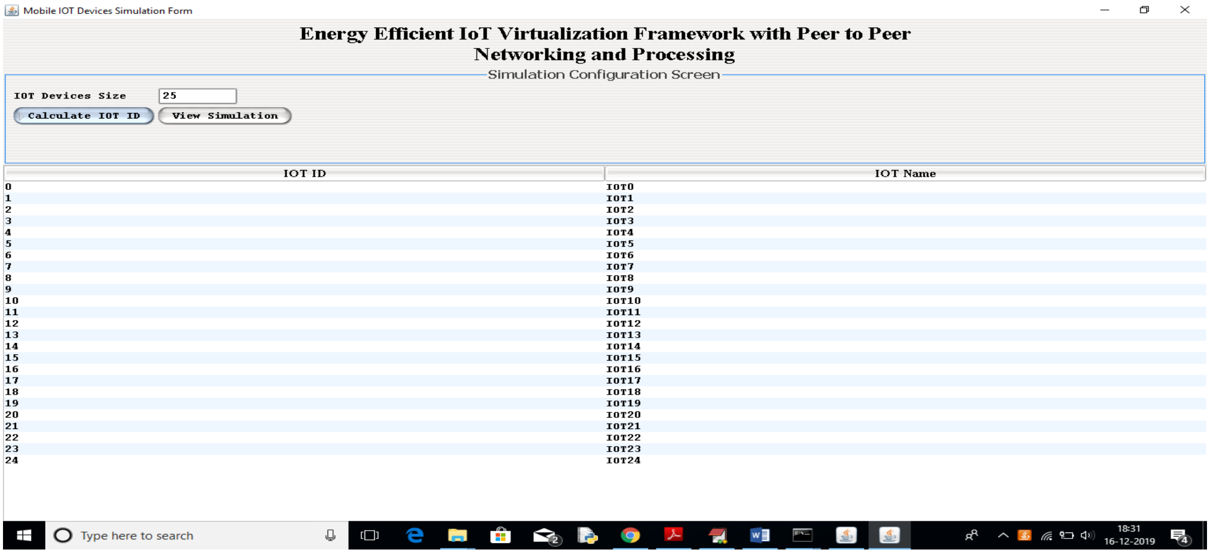


Fig: showing list of IOT devices

In above screen I entered IOT Device Size as ‘25’ and then click on ‘Calculate IOT ID’ button to assign ID to each device. Now click on ‘View Simulation’ button to get below screen

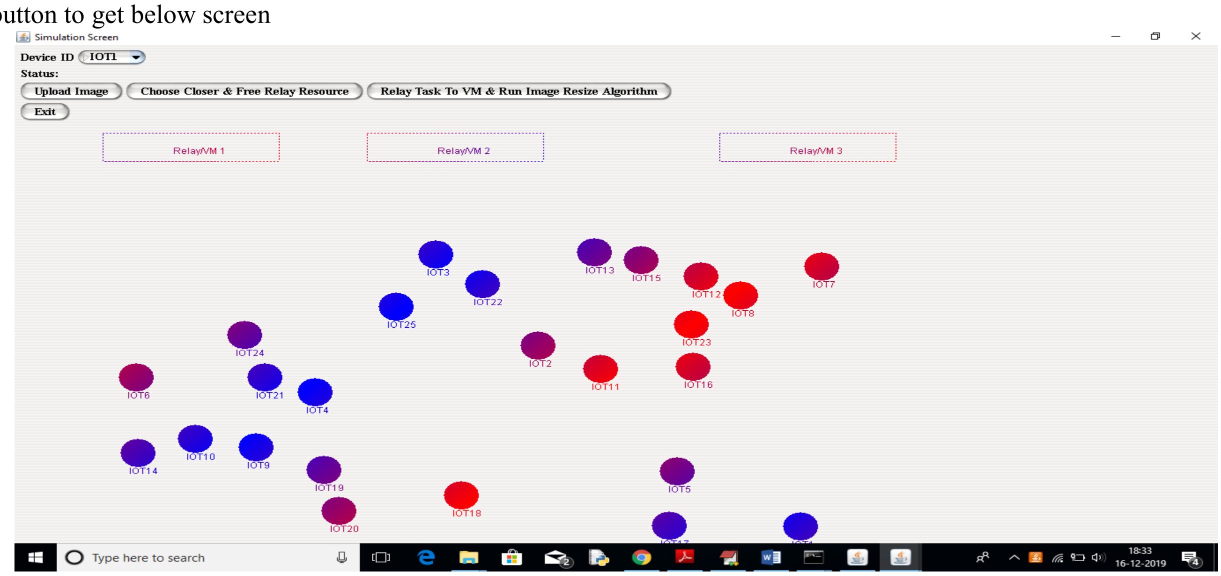


Fig: IOT devices

In above screen below Device ID we can see for selected IOT1 device closer VM is VM 3. Now click on ‘Relay Task to VM & Run Image Resize Algorithm’ button to enter new image height and width and then send that task to VM

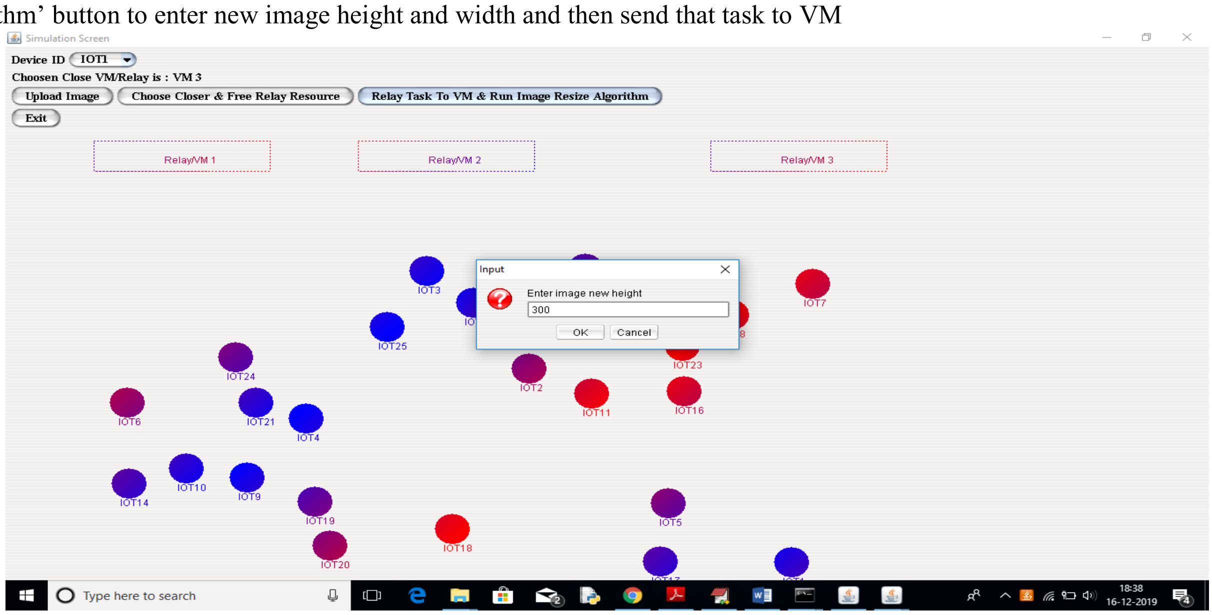


Fig: Uploading picture

In above screen entering image new height and width as 300 and 300. Now click ‘OK’ to offload task

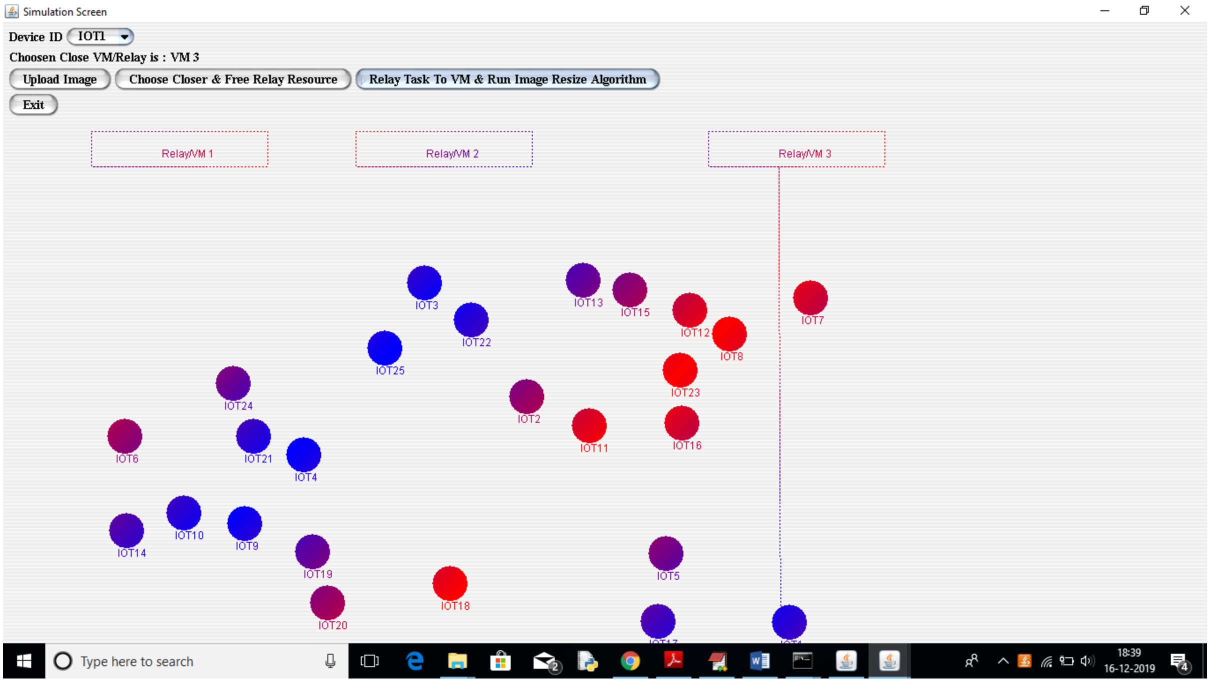


Fig: Choose closer and free relay resources

In above screen a line between IOT device and VM server indicates data is transferring between them and we get new resize image in below screen

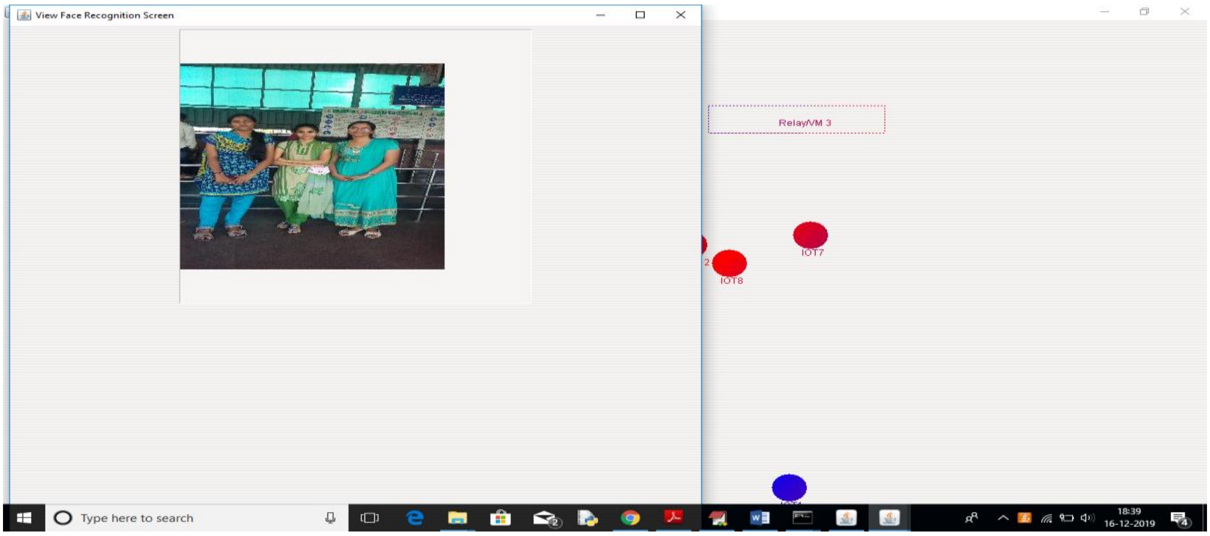


Fig: Selected image with new resolutions

Similarly you can select any IOT device and then upload image and resize to any size and test. Now we can see execution time graph at 'VM' screen below

In above screen we can see message that VM 3 received request to resize image for 300 and 300. Now click on 'Execution Time Graph' button in above screen to get below graph

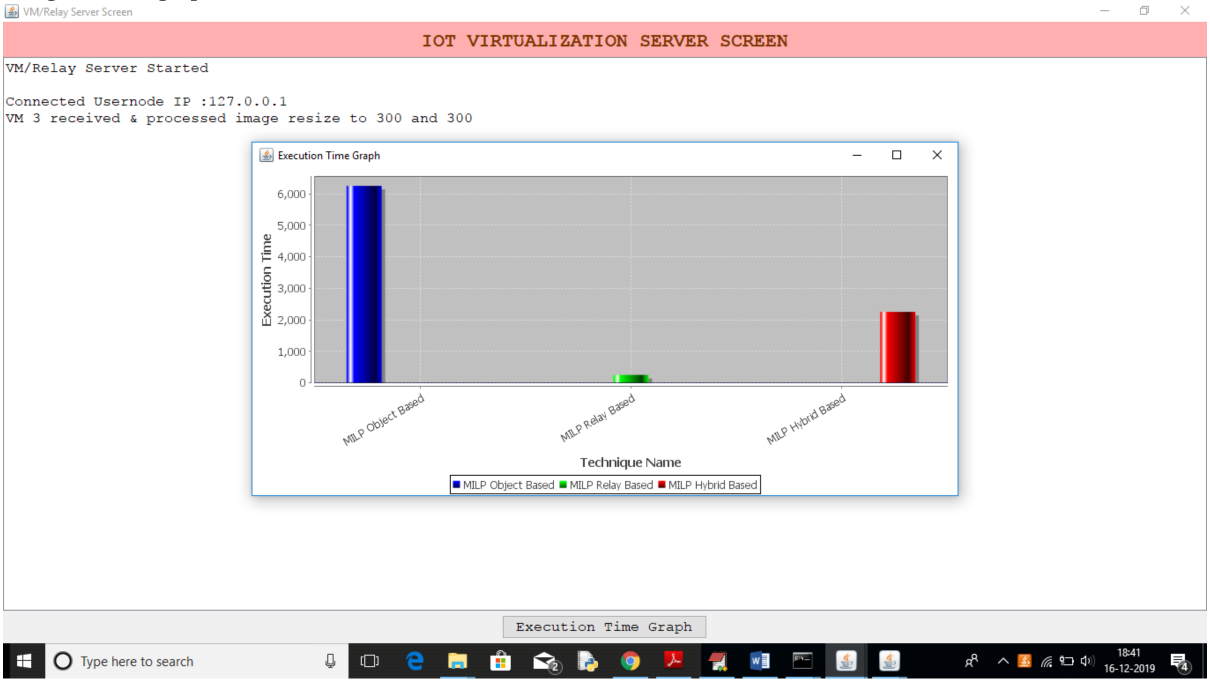


Fig: showing Graph

In above graph x-axis represents technique name and y-axis represents execution time and we can see MILP Relay technique took less execution time and its energy consumption will be less.

VI. Conclusion

In this paper, we have investigated the energy efficiency of an IoT virtualization framework with P2P network and edge computing. This investigation has been carried out by considering three different scenarios. A MILP was developed to maximize the number of processing tasks served by peers and minimize the total power consumption of the network. Our results show that the hybrid scenario serves up to 77% (57% on average) processed task requests, but with higher energy consumption compared with other scenarios. The relays only scenario can serve 74% (57% on average) of the processing task requests with 8% of power saving and 28% (22% on average) of task requests can be successfully handled by applying the objects only scenario with 62% power saving. The results also revealed the low percentage of addressed task requests in the objects only scenario resulting from the capacity limit of the IoT objects' processors. In addition, the small difference between the serving percentage of hybrid scenario and relays only scenario resulted from the allowed internal processing of objects in the hybrid scenario. For real time implementation, we have developed the EEVIPN heuristic based on the MILP model concepts. The heuristic achieved a comparable power efficiency and comparable number of executed tasks to the MILP model. The hybrid Scenario in the heuristic executes up to 74% of the total tasks (MILP 77%), up to 74% of tasks by the relays only scenario (MILP 74%) while the objects only scenario executes up to 21% of the tasks (MILP 28%). It should be noted that due to channel impairment and/or network congestion, link failures may occur, and hence retransmissions may become necessary. These retransmissions can have an impact on power consumption and therefore it is of interest to study the impact of resilience on energy consumption.

VII. Future Scope

In future, we can conduct additional studies by implementing other methods such as text mining and sentimental analysis to identify the linguistics of the content spread among the users in social networking sites by using data of the users like their language, locations, countries, preferences etc; can be involved for indepth study about information diffusion process.

References

[1] F. Ganz, D. Puschmann, P. Barnaghi, and F. Carrez, ``A practical evaluation of information processing and abstraction techniques for the Internet of Things," IEEE Internet Things J., vol. 2, no. 4, pp. 340354, Aug. 2015.

[2] J. Pan and J. McElhannon, ``Future edge cloud and edge computing for Internet of Things applications," IEEE Internet Things J., vol. 5, no. 1, pp. 439-449, Feb. 2018.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, ``Internet of Things: A survey on enabling technologies, protocols, and applications," IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2347-2376, 4th Quart., 2015.

[4] S. H. Shah and I. Yaqoob, ``A survey: Internet of Things (IOT) technologies, applications and challenges," in Proc. IEEE Smart Energy Grid Eng. (SEGE), Oshawa, ON, Canada, Aug. 2016, pp. 381-385.

[5] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, ``Energy ef- cient virtual network embedding for cloud networks," J. Lightw. Technol., vol. 33, no. 9, pp. 1828-1849, May 1, 2015.