EATSAL: An Energy Aware Task Scheduling Algorithm for Hybrid Networks

F. Hussain, A. Akram, M. Zafrullah

Abstract—**The widespread popularity of mobile computing devices, such as Laptops, handheld devices and cell phones, as well as recent advances in the wireless communication technologies have motivated researchers to provide novel solutions and applications for the users that were previously not feasible. The users of these mobile computing devices expect the same features and services from these devices as were previously available from conventional desktop computers. However to provide mobility and reduction of size of these mobile devices, the battery life is a major concern; several hardware based techniques have been proposed which results in more energy efficient systems as compared to the earlier systems. Even after these hardware improvement based techniques the problem still persists and it is believed that software based techniques have enough potential to reduce the energy demand to overcome the problems faced due to energy limitation.**

In this paper, we look into the problem of distributing the computational tasks among different devices in hybrid network environment. By hybrid networks we mean a network containing both wired as well as wireless handheld devices. The reason of selecting hybrid network environment is because most of the applications of mobile devices require accessing resources on the high bandwidth unlimited energy devices connected on wired network to help conserve the energy utilization of the energy limited wireless handheld devices. We have proposed a novel energy-aware scheduling algorithm to solve the problems of resource constrained mobile devices. Our scheduling algorithm schedules a set of computational tasks which may have operational and communication dependences, into the set of heterogeneous devices so as to minimize both the energy consumption and time taken by the tasks to be completed. Experiments show that significant improvement in the over all performance in terms of energy consumption and execution time of the handheld devices can be achieved by using our algorithm.

Index Terms— **Energy-aware Hybrid Networks, handheld devices, computational task, distributed scheduling, battery lifetime.**

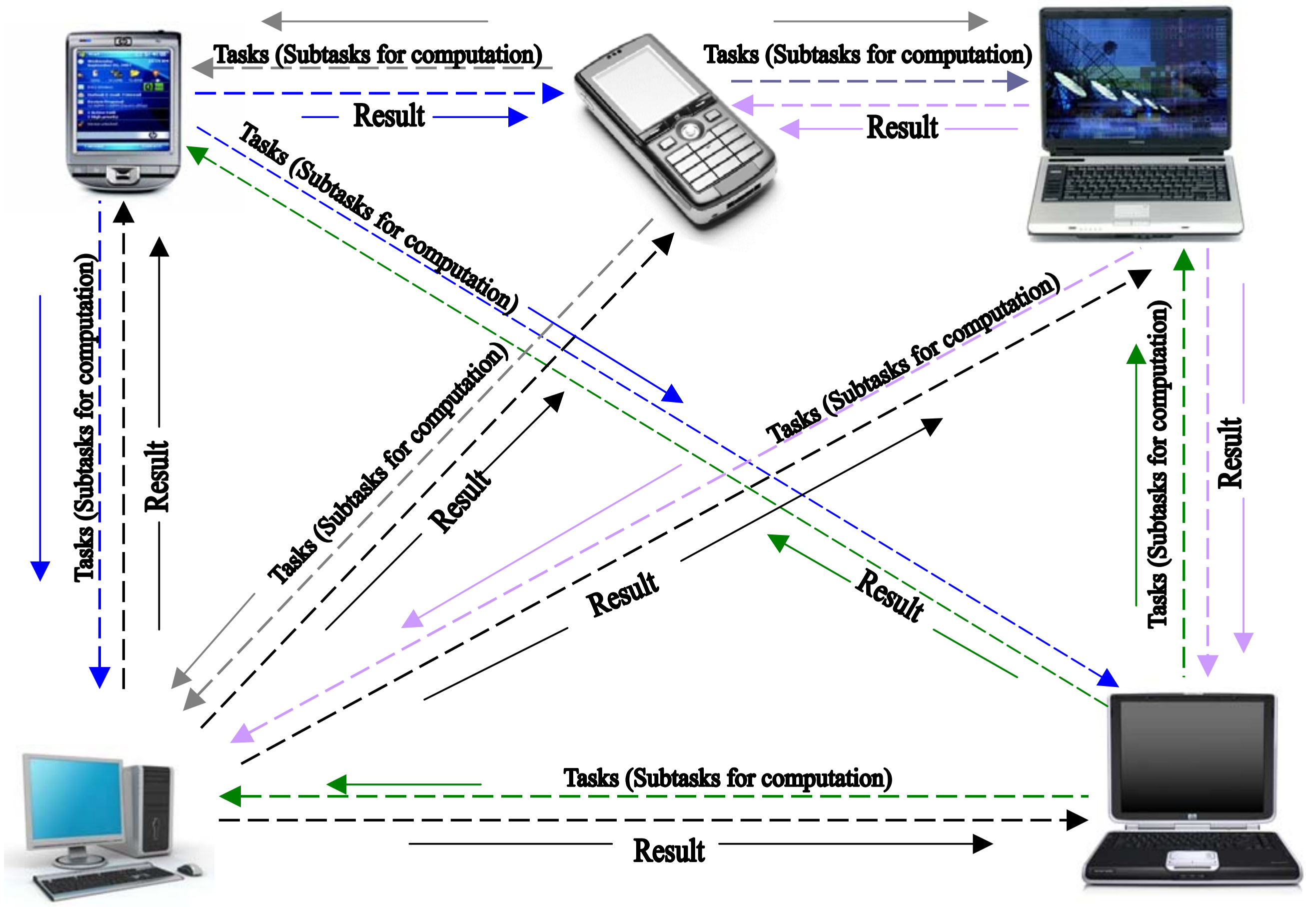
I. INTRODUCTION

Mobile computing devices and their applications are increasingly becoming popular due to their ease of use at anytime and any place. There is a range of applications from mobile commerce, geographical and location information, web services, streaming media and entertainment, etc. But in order to achieve the freedom of using these mobile applications on mobile devices one must be provided with the two pieces of technology: wireless communication facilities and mobile processing capabilities. While MANET's [1] provide the opportunity for anytime and any place (ubiquitous) communication structure, the mobile processing capabilities are still far away from being satisfactory [2]. The processing limitation is due to some constraints imposed by the mobility requirements of such devices. E.g. we require small and light weight components so that they are easy to carry with, but unfortunately this requirement imposes the limitation on processing capability and the energy (battery) capacity of mobile computing devices; energy has become a critical resource in such devices [3].

In order to improve the energy limitation problem, several hardware based techniques have been proposed and used for mobile computing devices. Turning off any idle component is one of the most common techniques [4] to conserve the energy of a mobile computing device. Voltage scaling is another technique of energy conservation i.e. some processors support the different voltage levels so the voltage usage is reduced which in turn helps in reducing the energy consumption by the processor [5][6]. Remote execution is software based technique in which the energy limited device transfers its computational task to another nearby device which is more powerful with respect to the energy constraint. While the computation offloading approach and hardware based approaches are orthogonal, the former has an advantage of improvement in both energy saving and processing time [7].

In this paper, we propose a cooperative computing approach for hybrid network environment that handles the problem of energy and a processing power limitation of mobile computing devices in the network, by hybrid network we mean that the network contains both wireless as well as wired computing devices. We claim that mobile computing devices involved in hybrid network environment can cooperate in running computational tasks in such a way that conserves energy and improves processing time through the deployment of remote execution platform (preferably wired devices) and the use of efficient-energy scheduler. Whenever a resource limited computing device (in such a hybrid network environment) has a set of tasks (or subtasks) to be computed (which may have operational and communication dependencies) it uses all the available resources in nearby computing devices (preferably wired and then wireless). Moreover the approach proposed in this paper is distributed (i.e. each device in the network has a scheduler running on it) so that it can easily decide a nearby device, to send a task (or a subtask) for computation based on its energy and computational capability. The scheduler plays the main role in this hybrid network environment by helping each device to find a proper schedule for task migration. Fig. 1 shows the architecture of our collaborative hybrid network environment in which all the devices are working in cooperation with all the other nearby devices. Moreover Fig. 2 shows the architecture of scheduler which is running on all the devices in the network, maintaining the required information, necessary for energy-aware task scheduling, of all the devices of in the network.

F. Hussain (fawad@uettaxila.edu.pk), A. Akram (adeel@uettaxila.edu.pk), M. Zafrullah (drzafrullah@uettaxila.edu.pk), all are working with Faculty of Telecommunication and Information Engineering, University of Engineering & Technology Taxila, 47050 Pakistan



Note:
In order to reduce the complexity in the above diagram, all the connections are not shown; otherwise each device is interacting with every other device for its tasks (subtasks) computation Based on energy and power of the device.

Fig. 1: Hybrid Network Environment

Distributed Energy-Aware Scheduler
1- Device id
2- Device Energy
3- Device Power
4- Tasks in queue

Fig.2 Energy-Aware Scheduler

II. RELATED WORK

Considerable amount of work in energy-aware scheduling targets the uni-processor architecture and aims at utilizing hardware based techniques to conserve energy of mobile devices, such as voltage scaling, switching off the device in idle state. Some work is also done on multiprocessor approach [2]. The author formulates a multi-processor energy aware scheduling problem for Mobile Adhoc Network (MANET's) and proposes a solution to the problem. The result shows an improvement in terms of energy conservation and execution time of the devices in the network, as compared to the earlier work. However the approach is somewhat different from ours in terms of the underlying architecture, characteristics of tasks to be scheduled (task subdivision included in our approach) and also our approach comes under the category of distributed scheduling because in this approach scheduler is running on all the devices in the network, either wired or wireless devices so that these devices are not relying on a single processor In terms of task to be scheduled for remote execution.

III. PROBLEM DEFINITION

In our work, we have introduced a novel non-preemptive distributed scheduling algorithm. The aim of this algorithm is to improve both the time and energy consumption of a mobile computing device and hence improve the overall performance of the device. Even though scheduling is a classical, common problem in many fields including computer science [8], the novelty of our scheduling problem is based on the availability of system resources to be utilized i.e. the battery energy, and processing power of the mobile computing devices so that they can perform for longer period of time in the network.

To define the task scheduling problem, one must define the task set and the processor set by the help of either a graphical and/or mathematical representation, mentioning all the characteristics of both the tasks set and the processors set in the network. Here it is important to note that the task subdivision is also considered in our algorithm so this must also be defined in the task set properties.

- Tasks: **T** = $\{t_1, t_2, t_3 \dots t_n\}$ is a set of tasks to be executed.
- Subtasks: Consider $\mathbf{t_1} = \{t_{1.1,} t_{1.2,} t_{1.3 \dots} t_{1.n}\}$ is a set of sub tasks of one big task (t_1 in this case) to be executed.
- Execution time: $\mathbf{T_{Texe}}$ represents the time taken by a task (or subtask) to be executed on any processor in the network.
- Energy consumption: $\mathbf{E_{Texe}}$**,** this represents the amount of energy consumed by a task (or subtask) to be executed.
- $\mathbf{E_{Tgen}}$, this represents the amount of energy consumed in task generation.
- $\mathbf{E_{Tdiv}}$ this represents the amount of energy consumed in task subdivision.
- $\mathbf{T_{Tgen}}$ this represents the amount of time consumed in task generation.
- $\mathbf{T_{Tdiv}}$ this represents the amount of time consumed in task subdivision
- Processors: P = $\{p_1, p_2, p_3, \dots p_n\}$ is a set of processors available in the system.
- Communication delay (Energy): $\mathbf{E_{Ttx}}$ and $\mathbf{E_{Trx}}$**,** this represents the amount of energy consumed by a task to transmit and received from one processor to the other respectively.
- Communication delay (Time): $\mathbf{T_{Texe}}$ and $\mathbf{T_{Ttx}}$**,** this represents the amount of time consumed by a task to transmit and received from one processor to the other respectively.

$$\mathbf{E_{ETask} = E_{Tgen} + E_{Tdiv} + E_{Ttx} + E_{Texe} + E_{Trx}} \quad \textbf{(1)}$$

Where, equation (1) is showing the amount of energy consumed by a task for generation, division, transmission, execution and reception back to the originating device.

$$\mathbf{E_{Dtotal} = \sum\nolimits^{n}_{i=1} E_{ETask(i)}} \quad \textbf{(2)}$$

Equation (2) is showing the total amount of energy consumed by any device to perform the above mentioned activities regarding all the tasks that it has generated of received by any other device in the network

$$\mathbf{T_{TTask} = T_{Tgen} + T_{Tdiv} + T_{Ttx} + T_{Texe} + T_{T\,rx}} \quad \textbf{(3)}$$

Equation (3) is showing the amount of time consumed by a task for generation, division, transmission, execution and reception back to the originating device.

$$\mathbf{T_{Dtotal} = \sum\nolimits^{n}_{i=1} T_{TTask(i)}} \quad \textbf{(4)}$$

Equation (4) is showing the total amount of time consumed by any device to perform the above mentioned activities regarding all the tasks that it has generated of received by any other device in the network.

IV. PROPOSED SOLUTION

As we have formulated the problem in the form of mathematical model in the previous section. In this section we will explain how our proposed scheduling algorithm is working. The important decision that is to be made here that how a device selects any processor in the network to compute the task that is originated in it. Our proposed algorithm is distributed, means that all the devices that are in the network (wired or wireless) each is having scheduler running on it, the scheduler in each device is maintaining the device identity, energy, power and the number of task queued for a device to be computed of all devices in the network. This is done by simply broadcasting a message by all the devices with a specified delay so that the data of each device is updated in the scheduler. After a specified amount of time the device list is reordered in the scheduler on the basis of a high energy device to be at the top
and after that the device which has the amount of energy remaining after the first one and so on.

Now whenever a any device originate a task, the device it self divide the task into the subtasks and then it selects the subtasks that it can compute itself, the remaining task for which the device has not enough energy to compute so then it looks for the other nearby devices to compute the task for it.

As explained earlier each device has a scheduler running on it which is maintaining the data regarding energy, power and number of tasks queued already for computation. So the device will select the top most processor in the list which have the maximum amount of energy and fewer tasks in queue. As we are working in collaborative hybrid network which has both wired and wireless devices so ideally every device tries to transmit its task (or subtask) to the wired (unlimited energy) device, but this is not the case always as that device might have tasks already in the queue to be computed so in that case a device has to select any other processor to compute the task but this decision is purely based primarily on the amount of energy of the device then power and number of task it already have in queue to compute. Here it is important to note that we are working in collaborative hybrid network, we have assumed that only the wireless (energy limited) devices are originating the tasks, wired (unlimited energy) devices are only in the network to help in computation of task and they are not originating any task.

The order of execution for the task is first come, first serve basis as our algorithm is non-preemptive so no task is assigned with any priority for execution.

A. Experimental Results

In order to prove that the energy-aware scheduling algorithm proposed by us is better in terms of both the energy consumption and time taken by the tasks to be computed, we conduct different several experiments using random task and processor set and compared the results of our energy-aware (EATSAL) scheduling algorithm (distributed) with one other energy-aware scheduling algorithm (centralized) proposed in [2]. The algorithm which is proposed in [2] considers that only one device is originating tasks and the scheduling algorithm is running only on that device and all other devices in the network are only there to help in task computation and that they are not at all originating any computational task. For the comparison purpose we have enhanced this algorithm in a sense that all the devices are originating computational tasks in them and they are helped by other devices to compute if they are not able to do themselves but the scheduler in running in one device which is managing the task transmission to the suitable device to compute and then sending back it to the originating device. It is important to note that the scheduling algorithm in [2] has assumed that task are already subdivided so no subdivision criteria is there, but in our proposed algorithm we have considered the task subdivision also, whenever any task originated in any device in the network, if a task is big and the device is not able to compute it itself then it simply subdivides the task and then transmit those subtask to the other nearby devices (preferably to wired and then wireless) for execution. It is also important to not that each subtask has an id attached which shows that of which device and a task it is part of.

Fig 1- 16 shows the randomly selected 15 tasks for different devices ,energy and time consumed for the computation of these tasks is shown for those individual devices. More over fig 17 and fig 18 shows the total energy and time consumed to compute the set of those randomly selected 15 tasks.

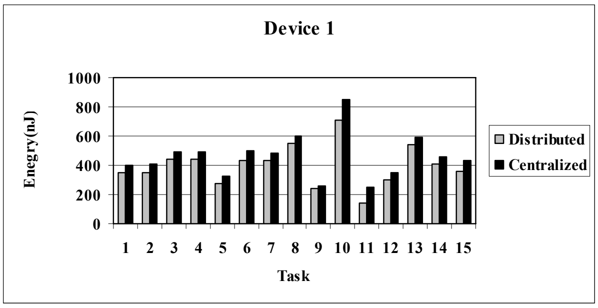


Fig. 1 Energy vs. Tasks (Device 1)

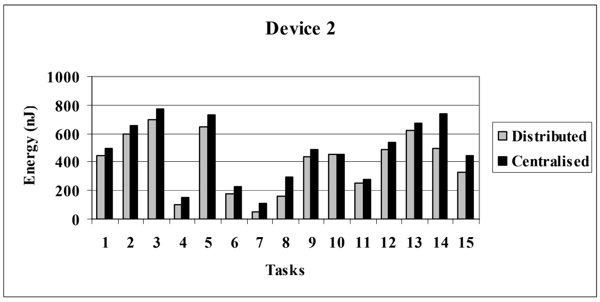


Fig. 2 Energy vs. Tasks (Device 2)

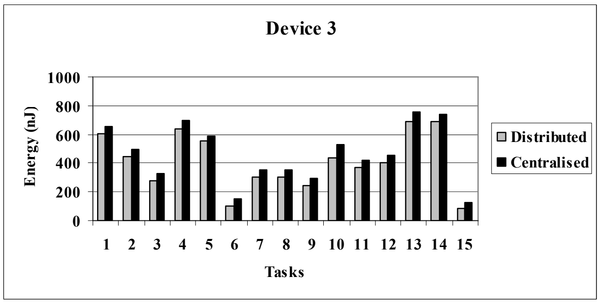


Fig. 3 Energy vs. Tasks (Device 3)

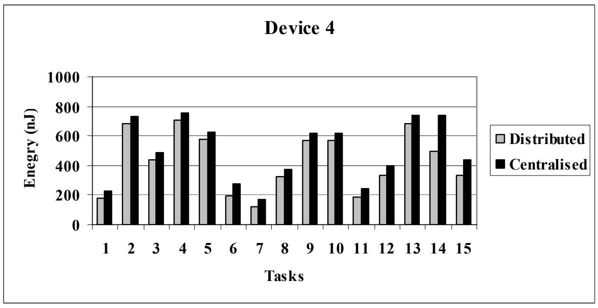


Fig. 4 Energy vs. Tasks (Device 4)

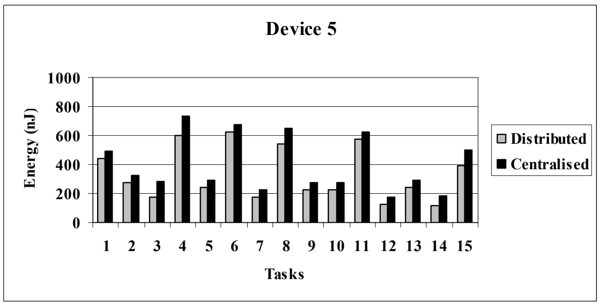


Fig. 5 Energy vs. Tasks (Device 5)

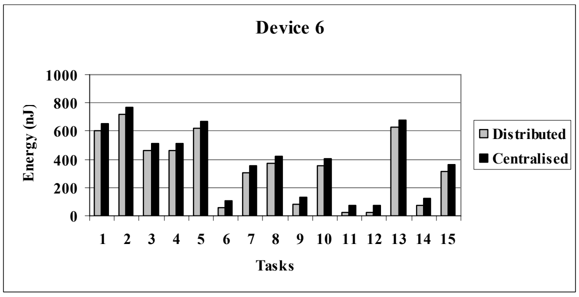


Fig. 6 Energy vs. Tasks (Device 6)

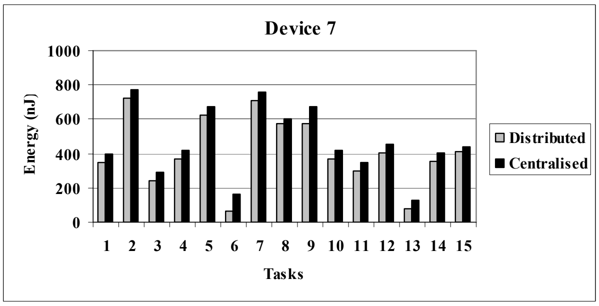


Fig. 7 Energy vs. Tasks (Device 7)

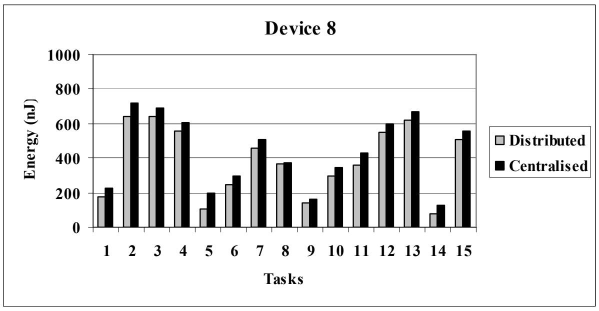


Fig. 8 Energy vs. Tasks (Device 8)

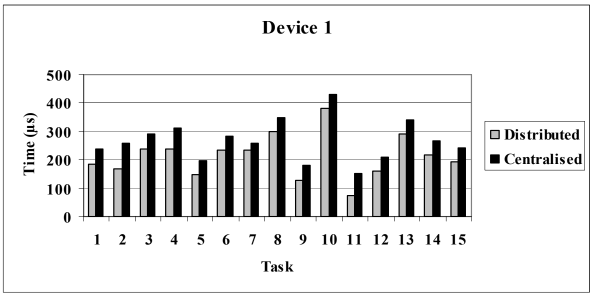


Fig. 9 Time vs. Tasks (Device 1)

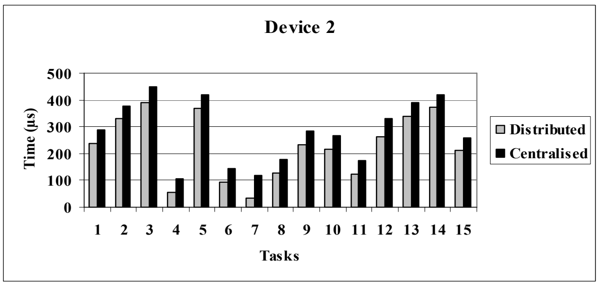


Fig. 10 Time vs. Tasks (Device 2)

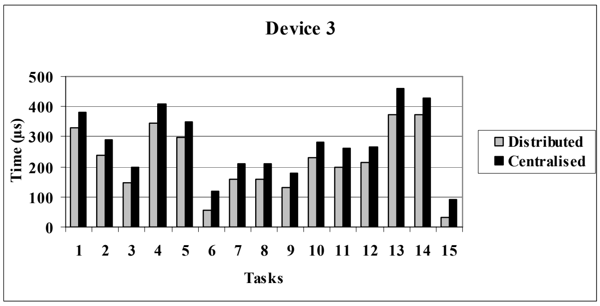


Fig. 11 Time vs. Tasks (Device 3)

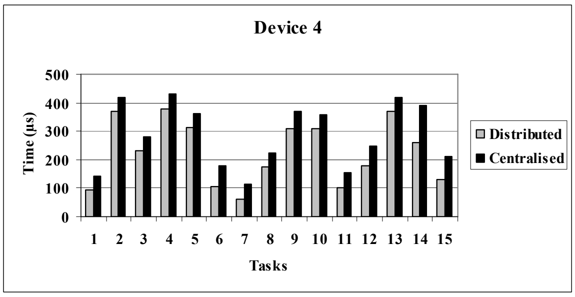


Fig. 12 Time vs. Tasks (Device 4)

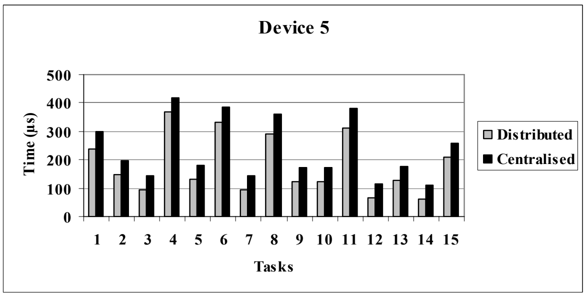


Fig. 13 Time vs. Tasks (Device 5)

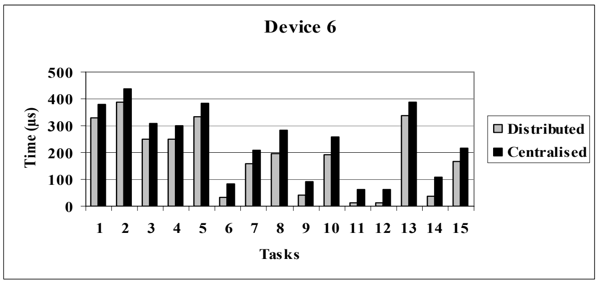


Fig. 14 Time vs. Tasks (Device 6)

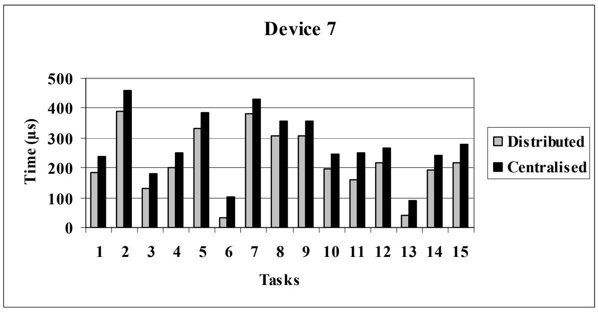


Fig. 15 Time vs. Tasks (Device 7)

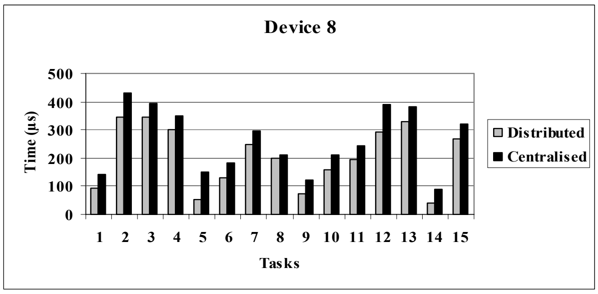


Fig. 16 Time vs. Tasks (Device 8)

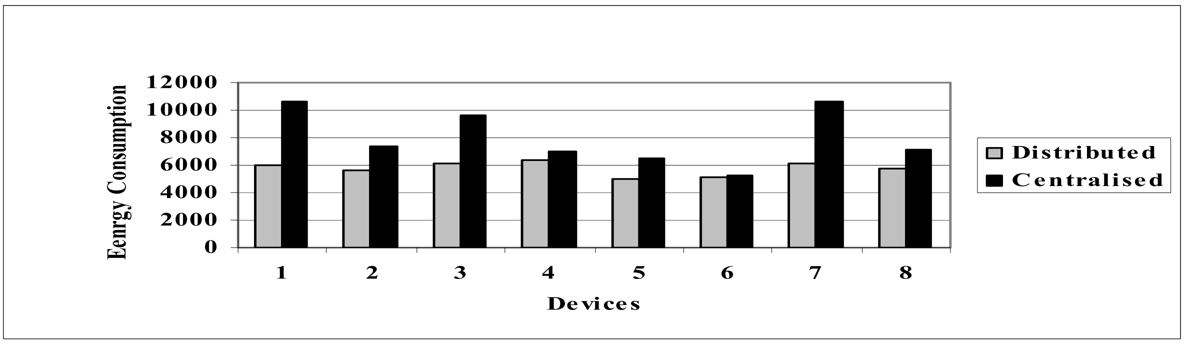


Fig. 17: Overall energy comparison

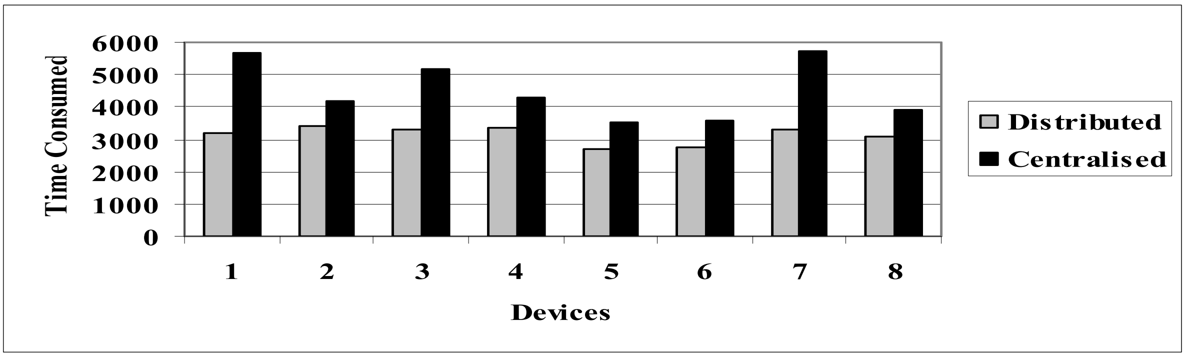


Fig. 18: Overall execution time comparison

V. CONLCUSION

Towards the high performance mobile computing, we have introduced distributed computing scheme over collaborative hybrid networks through the use of remote execution platforms. In this type of environment, the computational tasks of an energy limited device is distributed among the all other available devices (wired or wireless) in such a way by that it improves the performance of the device in terms of both the consumed energy and time taken to compute the tasks.

ACKNOWLEDGEMENT

First and foremost I would like to thank Mr. Naveed Khan Balouch for his continuous support and advice during my project work. Our interesting conversations and detailed discussions were fundamental to the success of this project.

I am grateful to Prof. Dr Adeel Akram, Chairman (TED) and Prof. Dr. M. Zafrullah, Dean (FT&IE) UET Taxila for their contribution to this work. Without their help this work would have never been possible. I am grateful beyond measures to my family, in particular to my parents for their encouragements. They have always kindly respected my decisions and supported me to achieve my goals.

REFRENCES

[1] C. Toh, “Adhoc Mobile Wireless Networks: Protocols and Systems”. PTR Prentice Hall, 2002.

[2] Alsalih, W.; Akl, S.; Hassanein, H.,”_Energy-aware task scheduling: towards enabling mobile computing over MANETs”, Proceedings of IEEE19th international Parallel and Distributed Processing symposium 2005

[3] H. Zeng, C. Ellis, A. Leveck and A. Vahdat, “Ecosystems: Managing energy as a first class operating system resource,” *In proceedings of Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, October 2002.

[4] K. Li, R. Kumpf, P. Horton and T. Anderson, “A Quantitative Analysis of Disk Driver Power Management in Portable Computers,” *In Proceedings of USENIX conference, pp. 279-292,* Winter 1994.

[5] R. Gonzalez, B. Gordon and M. Horowitz, “Supply and threshold voltage scaling for low power CMOS, “IEEE *Journal of Solid-State Circuits, 32, No 8. pp* 1210-1216, 1997.

[6] Jianli Zhuo; Chakrabarti, C, “An efficient dynamic task scheduling algorithm for battery powered DVS systems “Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific Volume 2, 18-21 Jan. 2005 Page(s):846 - 849 Vol. 2

[7] S. Gurun and C. Krintz, “Addressing the energy crisis in mobile computing with developing power aware software, “UCSB, Computer Science Department, Technical Report, 2003.

[8] E1-Rewini, T. Lewis and H. Ali, “Task Scheduling in Parallel and Distributed Systems”. PTR Prentice Hall, 1994.

[9] Olsen, A.B.; Fitzek, F.H.P.; Koch, P. “Energy Aware Computing in Cooperative Wireless Networks”, Proceedings of International conference on wireless networks, communication and mobile computing 2005.

[10] E. Coffman and R. Graham, “Optimal Scheduling of Two-Processor Systems,” *Acta Informatica, No 1, pp. 200-213,* 1972