https://dx.doi.org/10.4314/dujopas.v10i1c.1
ISSN (Print): 2476-8316
ISSN (Online): 2635-3490
Dutse Journal of Pure and Applied Sciences (DUJOPAS), *Vol. 10 No. 1c March 2024*

An Energy Conservation through an Adapted Probabilistic Scheduler for Timely Data Exchange in Local Mobile Cloud

Musa Muhammad[1*] and Modi B.[1]

Department of Computer Science,
Gombe state University

Email: musamuhammadjamari@gmail.com

Abstract

Mobile Cloud Computing has emerged as a pivotal technology, enabling mobile devices to harness external resources for hosting applications and significantly reducing latency. Recent research introduces the concept of a 'local mobile cloud,' formed by proximate mobile devices, to offload complex real-time applications to nearby devices, which minimizes energy requirement, and communication latency. This research introduces a more efficient task scheduling algorithm that is based on probabilistic task scheduling technique. This moves computations from multiple source nodes to closer processing nodes. A simulation model for local mobile clouds using OMNET++, is used for assessing the performance of the task scheduling algorithm. Additionally, a comparative analysis of the task scheduler with alternative scheduling schemes was conducted to evaluate performance in terms of the energy consumption, and process completion time. The outcome of the study showed that the probabilistic task scheduling technique improved the computing time and further conserved the energy resource requirement.

Keywords: Cloud computing; simulation; energy conservation; algorithm; Mobile Cloud Computing

INTRODUCTION

Mobile devices have indeed become an indispensable aspect of our modern daily lives. The year 2013 marked a significant milestone in the mobile industry, with over 1.99 billion mobile phones and tablets sold globally, as reported by analysts at Gartner (Gartner, 2020). This statistic underscores the widespread adoption and reliance on mobile devices in various aspects of our personal and professional lives.

In the field of mobile cloud computing, numerous studies have explored various aspects of task scheduling, resource allocation, and energy conservation. Afolayan et al. (2021) examined the opportunities and challenges of mobile cloud computing in Nigeria, highlighting the significance of leveraging cloud resources for mobile applications in resource-constrained environments. Cuervo et al. (2010) proposed MAUI, a system for offloading smartphone tasks to remote servers to conserve energy and extend battery life. Additionally, Nimmagadda et al. (2010) focused on real-time object recognition and tracking using computation offloading, demonstrating the potential benefits of offloading intensive tasks to cloud resources for timely processing.

**Author for Correspondence*

The continuous evolution of mobile devices, including improvements in CPU power, network connectivity, and the integration of advanced sensors, has expanded their utility far beyond mere communication tools. Today, people use their mobile devices for a wide array of tasks, ranging from essential activities like emailing and web browsing to entertainment pursuits such as gaming and multimedia consumption.
Mobile cloud computing, as demonstrated in Figure 1 is one way to get around these resource constraints by enabling the mobile device to transfer work to more capable resource devices, such as servers.

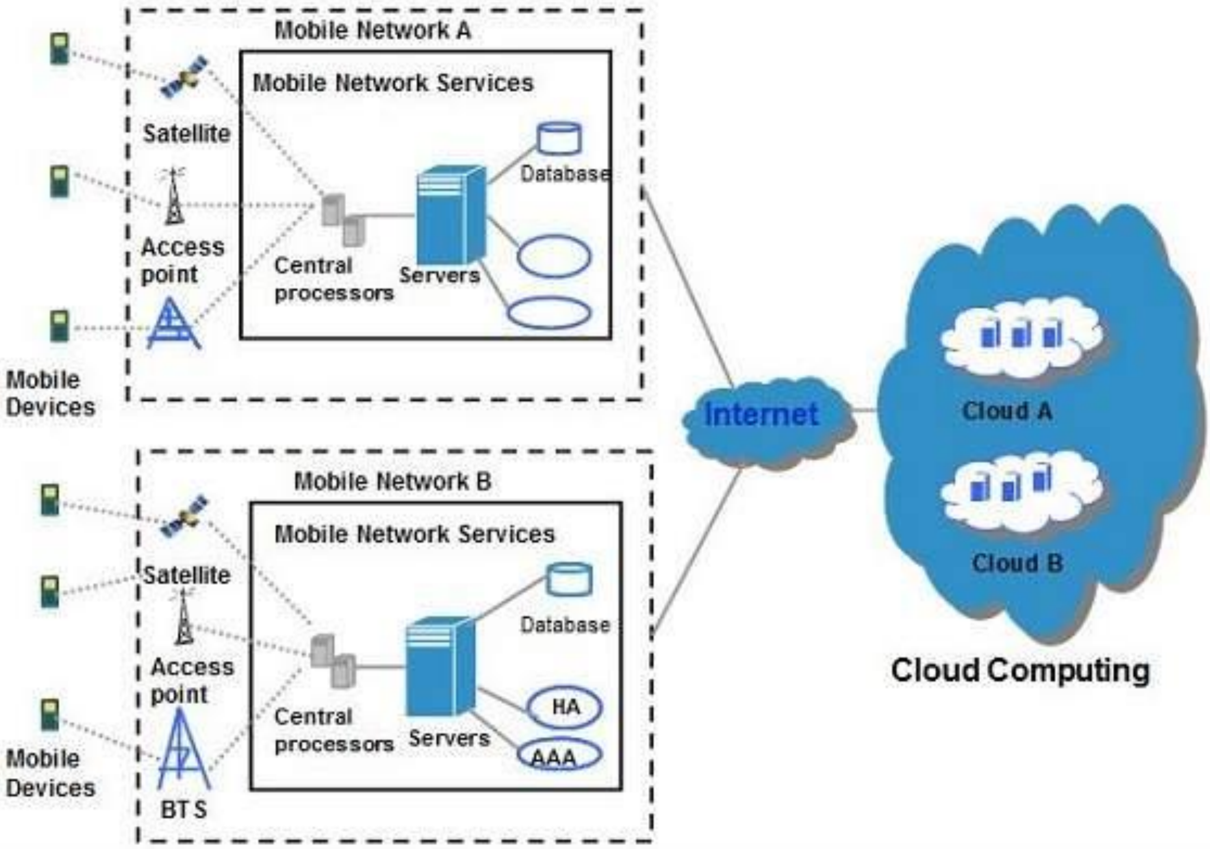


Figure 1: Mobile Cloud Computing Architecture. *Source*: Dinh et al. (2011).

Mobile Cloud Computing (MCC) combines mobile computing with five cloud computing technologies in a synergistic manner (McClatchey & Sanei, 2017). This paradigm allows mobile devices to interact with cloud infrastructure through robust internet connectivity, facilitating seamless offloading of resource-intensive tasks (Barbarani, Chiaramonte, & Corti, 2020). Offloading processes alleviate the burden on mobile devices, enabling complex computations and data processing in the cloud (Sultan, 2022). Notable advantages of MCC include cost efficiency and scalability, making it applicable in diverse domains such as mobile app development, augmented reality, and healthcare (Hassan, 2021). Heterogeneity in Mobile Cloud Computing presents taxonomy and open challenges (Sanaei, Abolfazli, Gani, & Buyya, 2019).

The research introduces a unique approach by developing an adapted probabilistic scheduler designed for timely data exchange within local mobile clouds. This scheduler integrates probabilistic principles into task scheduling and includes an adaptive mechanism to optimize performance in different network conditions. Unlike previous methods, this research offers a comprehensive solution addressing challenges in real-time application execution within local

mobile cloud environments. Factors such as energy conservation, completion time, and overhead costs are considered, enhancing the scheduler's efficiency and effectiveness.
The research fills a significant gap in the existing literature on mobile cloud computing by proposing an adapted probabilistic scheduler designed for energy-efficient task scheduling in local mobile clouds, integrating context-awareness and adaptation mechanisms, conducting comprehensive performance evaluations using OMNET++ simulations across various network scenarios, and demonstrating scalability and applicability, thereby advancing the state-of-the-art in energy-efficient task scheduling algorithms for real-time applications in dynamic mobile cloud environments.

MATERIALS AND METHODS

Notations and Assumptions

The research begins by establishing a set of notations and assumptions to define the foundation for modeling and algorithm development:

Notation	Description
V	Set of wireless nodes constituting the local mobile cloud.
E	Set of wireless links within the local mobile cloud.
(V,E)	Undirected topology graph representing the mobile cloud structure (V: node set, E: edge set).
μi	Average processing speed of node i in millions of instructions per second (MIPS).
Ei	Average energy consumption per million instructions of node i.
Ri	Radio transmission range of node i.
Bij	Bandwidth (bits per second) between nodes i and j.
tx$E$tx	Average energy consumption to transmit one byte.
rx$E$rx	Average energy consumption to receive one byte.
Qi	Queuing time experienced by node i.
J	Set of tasks arriving at node i.
Dj	Data size of task j.
Cj	Computation amount of task j in instructions.
Tj	Time constraint for task j.
τ	Time margin used when comparing estimated completion time with Tj.
P	Set of processing nodes in the local mobile cloud, $\subseteq P \subseteq V$.

Assumptions:

1. Tasks can be performed at any participating mobile device with processing power, need a lot of computing, and are not reliant on one another. One mobile device must be given a task to complete as soon as it arrives.
2. The size of the result returned from the destination is equal to the size of the data that has to be sent from the source node to the destination node.
3. Uniform transmission power is Adapted for the IEEE 802.11g communication protocol.
4. Every node inside the nearby mobile cloud is dispersed at random and linked through direct connections or ad hoc methods.

Methods

Literature Review: The study commenced with an extensive review of existing literature on mobile cloud computing, task scheduling, resource allocation, and energy conservation. This review served as the foundation for identifying gaps in current research.

Problem Formulation: Building upon the insights gleaned from the literature review, the study identified the pressing need for an energy-efficient task scheduling algorithm tailored for real-time applications within local mobile clouds. The objectives of the research were carefully defined, encompassing the development of a novel scheduling approach, the construction of a simulation model, the evaluation of performance metrics, and the execution of a comparative analysis against existing methodologies.

Algorithm Development: In response to the identified research gap, the study proposed an innovative task scheduling algorithm grounded in probabilistic principles and adaptive mechanisms. This adapted probabilistic scheduler was specifically designed to facilitate timely data exchange within local mobile clouds, aiming to optimize performance amidst varying network conditions.

Simulation Setup: To rigorously evaluate the performance of the proposed algorithm, the study leveraged OMNET++, a robust simulation framework. Through meticulous configuration of simulation parameters, including network topology, node density, task attributes, and communication parameters.

Experimental Evaluation: Comprehensive simulations were conducted within the OMNET++ environment to assess the efficacy of the proposed scheduling algorithm. These simulations involved evaluating various performance metrics, such as task completion rates and energy consumption, and comparing the results against those obtained from existing scheduling schemes. The aim was to provide empirical evidence of the algorithm's effectiveness in diverse network scenarios.

Result Analysis: Subsequent to the simulations, the study meticulously analyzed the obtained results to derive meaningful insights. By scrutinizing metrics such as task completion rates and energy consumption across different network configurations.

Adapted Scheduler

The resource discovery phase and the Adapted scheduling phase make up the two stages of the Adapted scheduler. Source nodes are able to obtain context information about neighboring processing nodes during the resource discovery phase. One processing node will be selected by the scheduler to carry out task j during the Adapted scheduling phase. Together, these two stages guarantee that local mobile cloud performance improves.

Phase I: Resource discovery phase

Based on QoS OLSR, the suggested resource discovery technique (Badis & Al Agha, 2020). Two types of control messages are available that provide resource information:

- *Modified Hello Messages,* which are sent locally (i.e. broadcasted to one-hop neighbors) to enable a node to discover its local neighborhood (as HELLO messages in the QoS OLSR protocol (Badis & Al Agha, 2020));
- *Modified Topology Control (TC) Message,* which are sent to the entire network through Multipoint Relay (MPR) nodes (Badis & Al Agha, 2020) to allow the distribution of the topology and context information to all the nodes (as TC messages in the QoS OLSR protocol (Badis & Al Agha, 2020)).

Keep in mind that periodic messages of both kinds are sent. The emission interval ought to be a function of the network's change rate. The emission interval should be shorter the faster the network changes. The emission intervals for the original Hello and TC massages are 2 and 5 seconds, respectively, according to Badis & Al Agha (2020). With the tight time constraints imposed by the applications, the Modified TC massages and Modified Hello Messages will have short emission intervals.

The direction of this research is to extend the routing table by including the following neighbor node parameters in the two different kinds of control messages:

- Device parameter: this one shows the node's processing speed and energy consumption percentage.
- Queue length: current queue time at node .

Algorithm 1 Resource Discovery Algorithm

Input: Control messages (TC or Hello messages).

Ouput: The table that holds the node parameter for every node is the output, or neighbor table. Function: At node u, handle control messages and update the neighbor table.

```
Procedure body:
{ initialize the neighbor table
listen control messages
if (Message Type== HELLO_MESSAGE)
{           update the neighbor table
if (node $u$ is an MPR node)
        construct/update TC message
 }
if (Message Type == TC_MESSAGE)
{
           update neighbor table
if (node u is an MPR node)
forward the TC message to all of its neighbors  }
}
```

Phase II: Adapted scheduling phase

Each time a source node u receives a task j submitted by its local user, it Estimates the energy consumption$Tasks\ Energy_{j,u,}$ and the completion time $Task\ Completion\ Time_{j,v}$ of task j on every potential processing node v ∈ P .

Algorithm 2 Adapted Scheduler

Input: Neighbor table from Phase I.,Task set J.

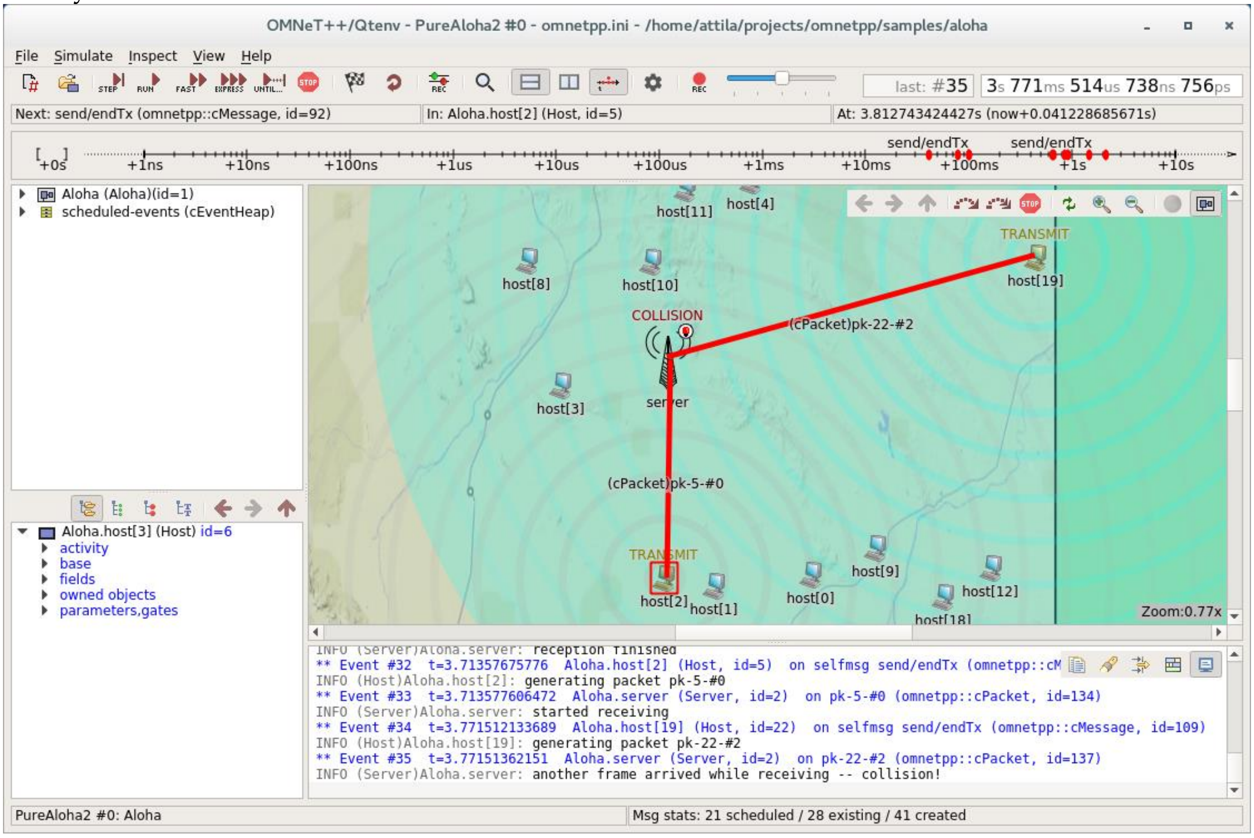
Output: Scheduling (Mapping).

```
for each processing node v in the neighbor table P:
{
  t_est = t_start + task_execution_time;
  E_task = computation_energy + communication_energy;

  if (t_est < T - t_start)
  {
    Add v to the set of eligible processing nodes P'.
  }
}

for each processing node v in P':
{
  p = CalculateProbability(E_task, v_parameters);
  w = RandomlySelectNode(P');

  Send task j to processing node w.
  Update estimated queue time on node w (tqw) as follows:
  tqw = t_start + t_est;
```

```
        Record the current time as t_complete.
        task_completion_time = t_complete - t_start.

        if (task_completion_time < T)
        {
          if (n_success > a)
          {
            T = T + Δt;
            n_success = 0;
          }
          else if (n_fail > a)
          {
            T = T - Δt;
            n_fail = 0;
          }
        }
      }
```

RESULTS AND DISCUSSION

Comprehensive simulations are carried out in OMNET++ to assess the performance of different scheduling strategies in local mobile clouds. For modeling and simulating computer networks, telecommunications systems, and other complex systems, OMNET++ is a popular open-source, modular, and component-based simulation framework. It offers a framework for developing, assessing, and verifying the functionality of different protocols, algorithms, and systems.



Next: send/endTx (omnetpp::cMessage, id=92)
In: Aloha.host[2] (Host, id=5)
At: 3.812743424427s (now+0.041228685671s)

```
INFO (Server)Aloha.server: reception finished
** Event #32  t=3.71357675776  Aloha.host[2] (Host, id=5)  on selfmsg send/endTx (omnetpp::cM
INFO (Host)Aloha.host[2]: generating packet pk-5-#0
** Event #33  t=3.713577606472  Aloha.server (Server, id=2)  on pk-5-#0 (omnetpp::cPacket, id=134)
INFO (Server)Aloha.server: started receiving
** Event #34  t=3.771512133689  Aloha.host[19] (Host, id=22)  on selfmsg send/endTx (omnetpp::cMessage, id=109)
INFO (Host)Aloha.host[19]: generating packet pk-22-#2
** Event #35  t=3.77151362151  Aloha.server (Server, id=2)  on pk-22-#2 (omnetpp::cPacket, id=137)
INFO (Server)Aloha.server: another frame arrived while receiving -- collision!
```

PureAloha2 #0: Aloha
Msg stats: 21 scheduled / 28 existing / 41 created

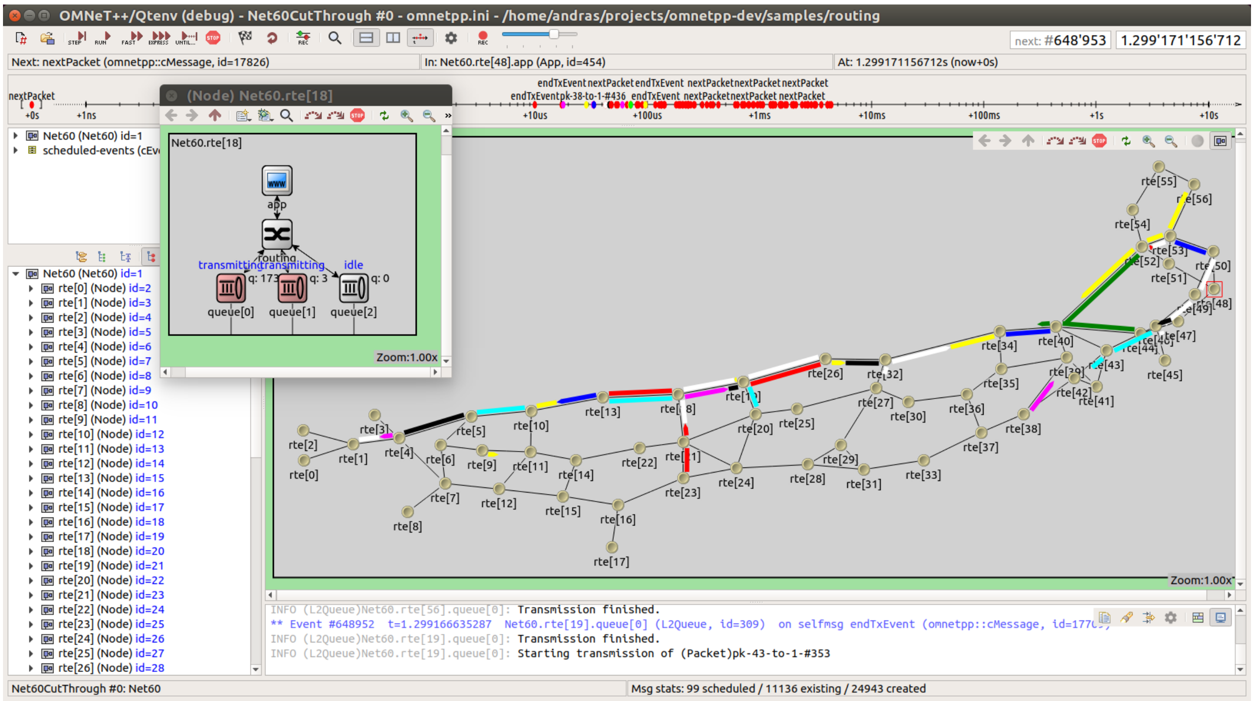
Figure 2: OMNET++ simulation runtime *Source*: Musa, M. (2023). An Energy Conservation through an Adapted Probabilistic Scheduler for Timely Data Exchange in Local Mobile Cloud. Unpublished Manuscript.

Simulation setup

A collection of nodes that can each send radio signals up to about 40 meters across an 11Mbps 802.11g wireless channel make up the simulated local mobile cloud. There are two simulated network situations, each with a distinct node density. In the first scenario, which is called the tiny network, ten nodes are randomly placed throughout a certain area. Two of these ten nodes are source nodes, and the remaining eight are processing nodes. Twenty nodes are placed at random throughout a region to generate the second scenario, often known as the huge network. Four of these twenty nodes are source nodes, and the remaining sixteen are processing nodes. Each source node generates a task. The event of the task arrival is a Poisson process. Parameters are listed in Table 1.

Table 1: Setting parameter *Source*: Musa, M. (2023). An Energy Conservation through an Adapted Probabilistic Scheduler for Timely Data Exchange in Local Mobile Cloud. Unpublished Manuscript.

Parameter	Value
Topology	Random
Network area	200m*200m
Network size	Small: 10 nodes / Large: 20 nodes
Communication range	Approximately 40 meters
Task data size	Varying from 1000 B to 8000 B
Task computation amount	Varying from 50 MI to 350 MI
Task time constraint	0.5s
Task arrival interval	Exponential distribution λ = 0.2s
Task arrival duration	200s
Computation ability of node *u*	Normal distribution in (1000,300) mips
Computation energy per MI of node *u*	$10^{-8}\times\text{mips}_u^2\text{J}$
Maximum bandwidth	11Mbps



Next: nextPacket (omnetpp::cMessage, id=17826)
In: Net60.rte[48].app (App, id=454)
At: 1.299171156712s (now+0s)

```
Net60 (Net60) id=1
scheduled-events (cEv
```

```
Net60 (Net60) id=1
rte[0] (Node) id=2
rte[1] (Node) id=3
rte[2] (Node) id=4
rte[3] (Node) id=5
rte[4] (Node) id=6
rte[5] (Node) id=7
rte[6] (Node) id=8
rte[7] (Node) id=9
rte[8] (Node) id=10
rte[9] (Node) id=11
rte[10] (Node) id=12
rte[11] (Node) id=13
rte[12] (Node) id=14
rte[13] (Node) id=15
rte[14] (Node) id=16
rte[15] (Node) id=17
rte[16] (Node) id=18
rte[17] (Node) id=19
rte[18] (Node) id=20
rte[19] (Node) id=21
rte[20] (Node) id=22
rte[21] (Node) id=23
rte[22] (Node) id=24
rte[23] (Node) id=25
rte[24] (Node) id=26
rte[25] (Node) id=27
rte[26] (Node) id=28
```

```
INFO (L2Queue)Net60.rte[56].queue[0]: Transmission finished.
** Event #648952  t=1.299166635287  Net60.rte[19].queue[0] (L2Queue, id=309)  on selfmsg endTxEvent (omnetpp::cMessage, id=177
INFO (L2Queue)Net60.rte[19].queue[0]: Transmission finished.
INFO (L2Queue)Net60.rte[19].queue[0]: Starting transmission of (Packet)pk-43-to-1-#353
```

Net60CutThrough #0: Net60
Msg stats: 99 scheduled / 11136 existing / 24943 created

Figure 3: Packet transmission *Source*: Musa, M. (2023). An Energy Conservation through an Adapted Probabilistic Scheduler for Timely Data Exchange in Local Mobile Cloud. Unpublished Manuscript.

The following metrics were used in the simulation studies to evaluate the suggested task scheduling scheme's performance to that of other algorithms.

Table 2: Comparative Result Analysis for Existing and Adapted Algorithms across two distinct networks, Network 1 and Network 2

Network 1			
Author's Name	**Scheduling Algorithms**	**Task Completion Rate(%)**	**Average Energy Per Successful Task(J)**
(Badis & Al Agha, 2020)	Round Robin	91	1.90
	Greedy	86	1.62
	Probabilistic	93	1.64
	Adapted Probabilistic	95	1.61
Network 2			
Author's Name	**Scheduling Algorithms**	**Task Completion Rate(%)**	**Average Energy Per Successful Task(J)**
(Badis & Al Agha, 2020),	Round Robin	90	2.24
	Greedy	77	1.93
	Probabilistic	88	1.76
	Adapted Probabilistic	89	1.68

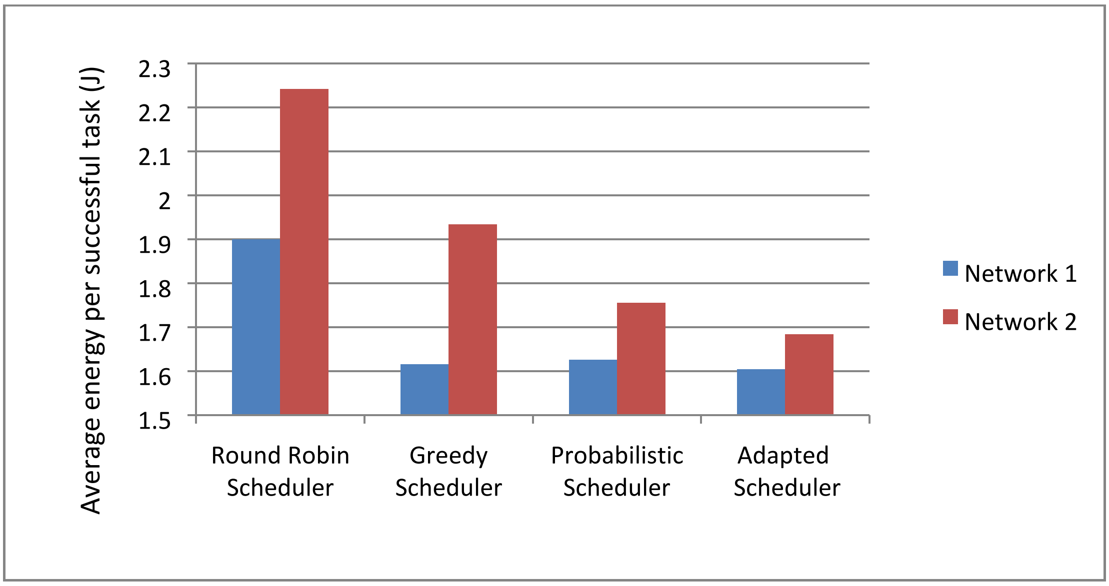


Figure 4: Average energy per successful task in small and large network *Source*: Musa, M. (2023). An Energy Conservation through an Adapted Probabilistic Scheduler for Timely Data Exchange in Local Mobile Cloud. Unpublished Manuscript.

In Figure 4. the average energy per task is 12.5% higher on average for all four schedulers in the large network. The main reasons are greater transmission energy and decreased completion rate. The advantage of the Adapted Probabilistic Scheduler in large network is more significant. Comparing with other schedulers, the Adapted Probabilistic Scheduler reduced 29.3% energy per task on average in the large network while 10.9% in the small network. The reason is that there are more energy efficient nodes in the large network. Thus the Adapted Probabilistic Scheduler can adjust its parameter to choose a target processing node from a larger set of processing nodes to avoid confliction.

Table 3: Comparative Result Analysis for Existing and Adapted Algorithms applied in two different scenarios: Stationary and Mobility

Stationary			
Author's Name	**Scheduling Algorithms**	**Task Completion Rate(%)**	**Average Energy Per Successful Task(J)**
(Badis & Al Agha, 2020)	Round Robin	0.91	2.15
	Greedy	0.89	1.15
	Probabilistic	0.91	1.19
	Adapted Probabilistic	0.93	1.15
Mobility			
Author's Name	**Scheduling Algorithms**	**Task Completion Rate(%)**	**Average Energy Per Successful Task(J)**
(Badis & Al Agha, 2020),	Round Robin	0.85	2.16
	Greedy	0.86	1.18
	Probabilistic	0.87	1.19
	Adapted Probabilistic	0.89	1.14

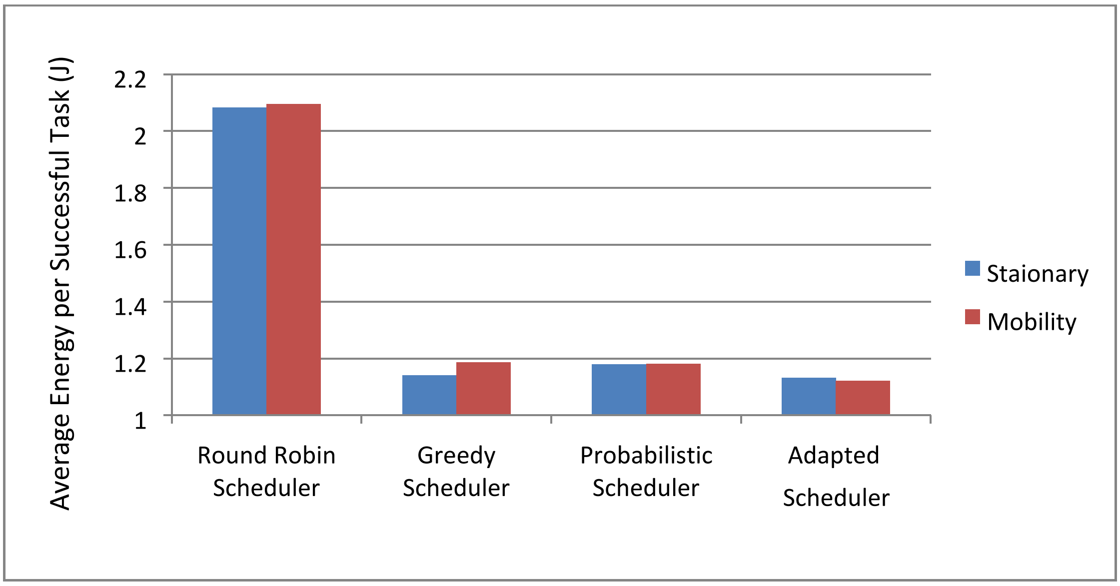


Figure 5: Average energy per successful task in stationary and mobile network *Source*: Musa, M. (2023). An Energy Conservation through an Adapted Probabilistic Scheduler for Timely Data Exchange in Local Mobile Cloud. Unpublished Manuscript.

Similar situation happens to the average energy per task. The round robin scheduler consumes 1.4% more energy in the mobile network than in the stationary network. The greedy scheduler consumes 6.57% more. The probabilistic scheduler consumes almost the same amount of energy in both networks. The Adapted Probabilistic Scheduler consumed 1% less energy in the mobile network. This result shows that the average energy per task is not heavily affected by the node mobility as to the task completion rate.

CONCLUSION

In real-time local mobile cloud applications, an adapted task scheduler integrates Quality of Service (QoS), Optimized Link State Routing (OLSR) to periodically update resource information. Leveraging enhanced QoS OLSR data, the scheduler calculates job completion time and energy consumption for potential processing nodes, utilizing a probabilistic scheduler to allocate tasks optimally. Additionally, it dynamically adjusts its time margin parameter to enhance performance in varying network conditions.

Overall, the experimental findings show that the Adapted scheduler can keep a high task completion rate while lowering the average energy required for each successful task. Furthermore, when the number of source nodes rises, the performance benefit becomes more pronounced. Furthermore, the suggested scheduler has the ability to adapt to operate in both fixed and mobile network settings. It also exhibits a high degree of adaptability to various task kinds. The Adapted scheduler's flexibility and scalability in local mobile cloud make it a potential solution for real-time applications.

REFERENCES

Afolayan, A., Adigun, M., & Mtenzi, F. (2021). Mobile cloud computing in Nigeria: Opportunities and challenges. International Journal of Cloud Applications and Computing (IJCAC), 8(2), 1-20.

Badis, H., & Al Agha, K. (2020). QOLSR, QoS routing for ad hoc wireless networks using OLSR. European Transactions on Telecommunications, 16(5), 427-442.

Barbarani, F., Chiaramonte, A., & Corti, D. (2020). A Cloud Based System for Mobile Augmented Reality Applications. Springer International Publishing.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2019). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 25(6), 599-616.

Cidon, A., London, T. M., Katti, S., Kozyrakis, C., & Rosenblum, M. (2020). MARS: Adaptive remote execution for multi-threaded mobile devices. In MobiHeld '11 Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, Article No. 1.

Cuervo, E., Balasubramanian, A., Cho, D. K., Wolman, A., Saroiu, S., Chandra, R., & Bahl, P. (2010). MAUI: Making smartphones last longer with code offload. Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, 49-62.

Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: Architecture, applications, and approaches. Wireless Communications and Mobile Computing, 13(18), 1587-1611.

Fantacci, R., Tarchi, D., & Tassi, A. (2010). A novel routing algorithm for mobile pervasive computing. In Global Telecommunication Conference, 1-5.

Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. Future Generation Computer Systems, 29(1), 84-106.

Hassan, Q. (2021). Mobile Cloud Computing: Challenges and Opportunities. Springer International Publishing.

Kemp, R., Yang, X., & van Renesse, R. (2019). eyeDentify: Multimedia Cyber Foraging from a Smartphone. In ISM '09 Proceedings of the 2019 11th IEEE International Symposium on Multimedia, 392-399.

Kumar, K., Liu, J., Lu, Y.-H., & Bhargava, B. (2013). A Survey of Computation Offloading for Mobile Systems. In Mobile Networks and Applications, 129-140.

Liu, X., Ranjan, R., & Madria, S. (2019). Energy-efficient task scheduling algorithms for cloud data centers: A survey. IEEE Communications Surveys & Tutorials, 21(1), 450-476.

Marinelli, E. E. (2019). Hyrax: Cloud computing on mobile devices using MapReduce. Master's Thesis, Carnegie Mellon University.

McClatchey, R., & Sanei, S. (2017). Mobile Cloud Computing: Foundations and Service Models. Wiley.

Nimmagadda, Y., Kumar, K., Lu, Y.-H., & Lee, C. S. G. (2010). Real-time Moving Object Recognition and Tracking Using Computation. In The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2449-2455.

Parekh, S. (1990). Analysis of a linear increase congestion control algorithm. In Proceedings of the conference on Communications architecture & protocols (pp. 66-75). ACM.

Sanaei, Z., Abolfazli, S., Gani, A., & Buyya, R. (2019). Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges. IEEE Communications Surveys & Tutorials, 16(1), 369-392.

Shi, S., Hsu, C.-H., Nahrstedt, K., & Campbell, R. (2020). Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming. In MM '11 Proceedings of the 19th ACM International Conference on Multimedia, 103-112.

Sultan, N. (2022). Mobile Cloud Computing: A Review. Future Generation Computer Systems.

Verbelen, T., Simoens, P., De Turck, F., & Dhoedt, B. (2012). Cloudlets: Bringing the Cloud to the Mobile User. In Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services, 29-36.

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1), 7-18.