Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

NETWORK AWARE RESOURCE SCHEDULING IN SENSOR CLOUD

SAYED CHHATTAN SHAH
Department of Information Communication Engineering
Hankuk University of Foreign Studies, SOUTH KOREA
E-mail: shah@hufs.ac.kr

ABSTRACT

In sensor cloud, multiple sensing and computing devices interconnected through an ad hoc network are presented as one powerful unified computing resource. One of the key issues in sensor cloud is resource scheduling which is concerned with identification of computing resources where the application tasks will be run. In literature, several resource scheduling schemes have been proposed but most of them are either targeted towards pre-existing network infrastructure-based systems or they do not consider the dynamic and distributed nature of sensor cloud. In this paper, an energy efficient and network aware resource scheduling scheme is proposed for scheduling of data intensive tasks on sensor cloud. The scheme uses multi-level transmission power and network information to reduce transmission energy consumption and data transfer time.

Keywords: *Sensor grid, Mobile cloud computing, Ad hoc sensor network, Task scheduling, Energy-aware*

1. INTRODUCTION

In battlefield, soldiers may experience physical and mental problems. In such situations, various biomedical sensing devices, capable of acquiring vital signs such as blood pressure and flow, temperature, electro cardiogram, oxygen saturation, and CO2 concentration, can be used to continuously monitor the soldiers' psychophysiological health. The data generated by biomedical sensing devices then can be exploited 1) to assess the physical and mental stress and associate it with the current and past activities of the soldier, and 2) to perform rapid trauma triage in case of injuries. In addition, soldiers also need to rely on various sensing, processing and communication systems in the vicinity to achieve situational awareness and understanding of the battlefield. However, simultaneously executing computationally intensive models [1-4] for deriving physiological parameters from vital signs and for acquiring context and battlefield awareness in real time requires computing capabilities that go beyond those of an individual sensing and processing devices. In order to execute computationally intensive models, a sensor cloud is proposed.

A sensor cloud enables mobile computing and sensing devices to execute computationally intensive tasks in ad hoc environment. It consists of multiple devices interconnected though an ad hoc network. Interconnected devices are dynamically provisioned and presented as one powerful unified computing resource.

The sensor cloud is the integration of cloud, sensors and ad hoc network. The cloud enables convenient and on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Whereas ad hoc network is a wireless network of mobile devices that communicate with each other without any pre-existing network infrastructure [2].

One of the key issues in sensor cloud is resource scheduling which is concerned with identification of computing devices where the application tasks will be run. Compared to traditional cloud computing systems, resource scheduling in sensor cloud is difficult due to low bandwidth, limited battery power, and dynamic and infrastructure-less communication environment [1] [38].

In this paper, an energy efficient and network aware resource scheduling scheme is proposed for scheduling of data intensive tasks on sensor cloud. The scheme uses multi-level transmission power and network information to reduce transmission energy consumption and data transfer time. This

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

paper also investigates the relationship between data transfer time and transmission energy consumption.

The rest of the paper is structured as follows. Section 2 discusses related works and Section 3 presents the system architecture. Resource allocation schemes are presented in Section 4. In Section 5, simulation results are discussed. Section 6 presents the conclusions to the paper.

2. RELATED WORK

A hybrid static and mobile grid computing system is proposed in [32] in which mobile and static computing devices and bio-sensing nodes are integrated and presented as a one unified system. The bio-sensors collect vital signs such as blood pressure, temperature, electrocardiogram, and oxygen saturation of an individual. The collected data is processed and analyzed on mobile grid computing infrastructure in order to determine the health of an individual. To deal with uncertainty, an idea of application waypoints has been introduced in which service provider executing application task reports to the broker with an estimate of residual task completion time. If the broker does not receive feedback about the estimated residual task completion time from the service provides at the specified waypoint, it marks service provider as failed and assigns additional resources to take over the incomplete tasks. A resource allocation algorithm to efficiently process telemedicine data in the grid is proposed in [37]. In proposed algorithm sensors attached to patient's body collect and send health related data to grid through a mobile device. A patient management application deployed on the grid processes and analysis the patient's data.

A sensor-cloud infrastructure proposed in [31] integrates sensors with cloud computing system. In sensor-cloud infrastructure, physical sensors integrated with cloud computing system are virtualized as virtual sensors and are provided as a service. A pull-based resource allocation algorithm is proposed in [33] in which a service provider node pulls tasks from the service broker nodes, executes them and submits results once task completes its execution. In [34] authors have proposed a sensor grid platform to combine real-time data about the environment with vast computational resources. The proposed sensor grid platform can be deployed using centralized architecture or decentralized architecture. In centralized architecture, a sensor network connected to grid collects data while processing of data is carried out on the grid. In distributed architecture, a sensor network collects data and performs simple data processing tasks while computationally intensive data processing and analysis tasks are executed on the grid.

A sensor data collection network to integrate sensor data into grid applications is discussed in [35]. The sensor data collection network includes three key components: the data collection network, sensor entry points, and application entry points. The data collection network discovers, filters, and queries multiple sensor networks. Each sensor network has one or more sensor entry points that map application data requirements onto low-level sensor network operations. The application entry points provide application connectivity to data collection network. The proposed system is based on publish-subscribe paradigm. A sensor network publishes sensor data and metadata through service entry points while application subscribes to sensor network and receives a data in real time. In [36] authors have proposed a scalable proxy-based architecture for sensor grid in which nodes in wireless sensor network are provided as a service on the grid. The use of proxy-based architecture where proxy acts as an interface between grid and sensor network supports a wide range of sensor network implementations. The systems and schemes such as [9-19] and [31-37] assume pre-existing network infrastructure and therefore are not suitable for ad hoc and dynamic network environments.

The schemes proposed for ad hoc and dynamic network environments do not consider dependencies between tasks and connection quality between nodes. In addition, they are targeted towards processing energy consumption, load balancing and fault tolerance rather than application performance and energy-efficient communication between the tasks.

Compared to existing work, proposed scheme considers task dependencies and aims to reduce transmission energy consumption and application completion time. It uses multi-level transmission power and network information for decision making.

3. SYSTEM MODELS

A sensor cloud comprises a plurality of mobile computing and sensing devices communicating through an ad hoc network. Each node uses multiple transmission power levels. A sensor cloud

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

may include various types of devices such as audio and video sensors, smart mobile and stationary robots, and smart phones. An application consists of independent and interdependent tasks divided into three categories: computation-bound tasks, local communication-bound tasks, and remote communication-bound tasks [4].

4. SCHEDULING IN SENSOR CLOUD

To schedule tasks there are three cases: (a) scheduling of an independent task, (b) scheduling of an interdependent task set and (c) scheduling of tasks that have a dependency with already scheduled task.

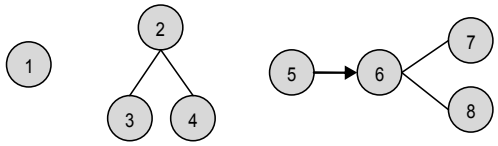

Figure 1: Three possible cases: a) task 1 is independent, b) tasks 2, 3 and 4 form an interdependent task set. Task 2 has parallel execution dependency with task 3 and 4 and vice versa, and c) task 5 is independent task while tasks 6, 7 and 8 form an interdependent tasks set. The task 6 in a set has a precedence dependency with task 5.

This paper addresses the problem of scheduling interdependent tasks set, which consists of tasks with parallel execution dependencies [5]. To schedule interdependent tasks set to closely located nodes there is need to search a group of closest nodes within the sensor cloud.

The problem of searching a group of closest nodes is modelled as a k-nearest neighbor (kNN) search problem in graph theory [5].

To discover kNN, every node broadcasts a discovery message at a minimum transmission power level. If the number of discovered nodes >= k the search process stops and the discovered nodes are declared as kNN. After the discovery process, each node calculates weight using equation 1.

$$n_i^w = \sum_{i=1}^{l} TPL_i^w . N_{TPL_i} \qquad (1)$$

TPL_i^w = weight of TPL_i

n_i^w = weight of node n_i

l = number of transmission power levels

N_{TPL_i} = number of nodes accessible at TPL_i

4.1 Power-based Resource Scheduling

The power-based resource scheduling scheme schedules tasks on nodes reachable at minimum transmission power level. To schedule tasks set, each node uses kNN algorithm to discover KNN and calculate weight. The weigh is sent to resource scheduler, which schedules task on a node with highest weight. The reader is referred to [4] for a detailed description of the kNN search algorithm and power based resource scheduling.

4.2 Network-aware Resource Scheduling

In sensor cloud the bandwidth at different network portions fluctuates over the time and different nodes often experience different connection quality at the same time. Since connection quality between nodes is not same and varies over the time, an effective and efficient resource scheduling scheme should consider connection quality between nodes in addition to other factors such as processing speed, transmission power and distance between nodes.

Connection Quality

The connection quality is measured in terms of round trip time, packet loss probability and medium access delay.

$$C_j = (P_{size} / T_{ready} - T_{recieved}) * P_{loss} \qquad (2)$$

C_j = Connection quality

P_{size} = Packet Size

T_{ready} = time packet is ready to be transmitted

$T_{recieved}$ = time acknowledgement received

P_{loss} = Packet loss ratio

Similar metrics have also been used in [14] and [16]. They only differ with respect to number of parameters taken into consideration. For example, a scheme proposed in [14] does not consider time to gain access to medium and also demonstrate that medium access delay does not have significant impact on the performance. The experiments were conducted with medium network size and moderate network load. On other hand, schemes such as [15] and [16] have reported that medium access delay plays a significant role as number of nodes and data transfers increase within a network.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Resource Scheduling Scheme

Each node manages connection quality data of kNN and based on this calculates rank. Resource scheduler schedules tasks set on a node with highest rank. The selected node receives tasks set and schedules tasks in a set to kNN based on CPU speed and task information. The rank is calculated using equation 4.

$$R_i^{total} = \sum_{j=1}^{k} C_j \quad (3)$$

$$n_i^r = R_i^{total} / k \quad (4)$$

C_j = Connection quality between node n_i and n_j

n_i^r = rank of node n_i

k = Number of nearest neighbors

4.3 Power-based versus Network-aware Resource Scheduling

In contrast to power based resource scheduling, network aware resource scheduling scheme may increase transmission energy consumption and network capacity. There is a trade-off between power based scheduling and network aware scheduling [5].

4.4 Energy-efficient and Network-aware Resource Scheduling

The weight calculated based on transmission power and rank calculated based on communication quality are used to calculate grade of a node. The weight reflects transmission energy consumption and network capacity whereas rank reflects data transfer time. Grade of a node is viewed as an attempt to balance between weight and rank.

$$n_i^g = (1-\alpha) * n_i^w + \alpha * n_i^r \quad (5)$$

n_i^r = rank of node n_i

n_i^g = grade of node n_i

n_i^w = weight of node n_i

α = tunable parameter subject to $0 \le \alpha \le 1$

To allocate an interdependent tasks set, the resource scheduling service first sends the m to member nodes, where m is the number of tasks in a set. Each member runs the kNN search algorithm, where k >= m, calculates the grade and sends the grade to the resource scheduling service. The resource scheduling service then selects the node with the highest grade. The decision to allocate tasks within a set is made by the selected node, which then allocates the tasks to its kNN according to task type.

SUMMARY

- Send m to member nodes, where m is number of tasks in a set.
- Each node check if kNN >= m
- If kNN are not discovered yet or kNN<m,
  - Start kNN search process
  - Calculate
    - Weight using equation 1
    - Rank using equation 4
    - Grade using equation 5
  - Send grade to resource scheduling service
- Resource scheduling service select a node with a highest grade, which then allocates tasks in a set to its k nearest neighbors

5. SIMULATION RESULTS

The performance of proposed scheme (ENRA) is compared with power-based resource scheduling scheme (PRA) [4] and network aware rank-based resource scheduling scheme (NRA) [5]. Power-based resource scheduling scheme allocates interdependent tasks to nodes based on transmission power control mechanism while network aware rank-based scheduling scheme uses network information in order to make allocation decisions.

5.1 Performance Metrics

Accumulative application completion time

$$A_{CompTime} = \sum_{i}^{n} T_{CompTime}^{i}$$

Task completion time

$T_{CompTime}$ is defined as a time that task takes to complete its execution.

Energy Consumption

Energy consumed in transmission of data.

5.2 Simulation Setup

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The network simulator NS2 was used for performance evaluation with a wide range of scenarios.

Table 1: Simulation Parameters

Simulation Time	7000 seconds
Number of Nodes	15-20
Transmission Power Levels	2-5
Transmission Range	90m-130m-170m-210m-250m
Simulation Area	1500m X 1500m
Number of Tasks	10-20-30-40
Transport Protocol	TCP
Ad Hoc Routing Protocol	ExClusterPOW
MAC Protocol	IEEE 802.11
Packet Size	512 Bytes
Value of tunable parameter α	0.5

In order to make allocation decisions, three key services were implemented: monitoring service, discovery service and resource scheduling service. Monitoring service runs on nodes willing to share computing resources while resource scheduling and discovery services execute on node that requires additional computing resources. The parameters specific to scenarios are described in respective sections while simulation parameters are given in Table 1.

5.3 Simulation Results

The experiments are performed to evaluate the performance of the proposed scheme in various scenarios with respect to the network and the application configuration. The amount of data transfer between tasks is varied to reflect a range of applications such as automated video surveillance, distributed object tracking and 3-D scene construction. To evaluate the performance we set 6 different scenarios using 3 different network setups and 2 applications as follows:

Table 2: Network Setup

	Network setup 1	Network setup 2	Network setup 3
Number of nodes	15	20	20
Groups' size	Variable	Variable	Fixed
Maximum number of nodes in group	6	3	4
Min number of nodes in group	2	2	4
Distance between nodes in groups	20-50m	40-50m	50-100m
Distance between nodes across groups	100-250m	150-250m	100-250m
Number of nearest neighbors	3	3	3
Transmission power levels	3	3	2

Table 3: Application Configuration

	Application 1	Application 2
Max number of interdependent tasks in set	3	4
Min number of interdependent tasks in set	2	3
Data transfer size for each task	4-8 *MB*	4-8 *MB*
Inter-packet interval	10-300ms	10-100ms
Tasks arrival order	Order 1	Order 2
Ratio of interdependent task set of max size to interdependent task set of min size	1:1	1:2

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: Scenarios Setup

	Network Setup	Application	Data transfers before allocation process
Scenario 1	1	1	*
Scenario 2	2	1	*
Scenario 3	3	1	*
Scenario 4	1	2	*
Scenario 5	2	2	*
Scenario 6	3	2	+

* *UDP based Constant bit rate applications were started*

\+ *TCP based Constant bit rate applications were started*

5.3.1 Accumulative Application Completion Time

Figures 2-4 demonstrate an AACT for scenarios 1-3. NRA improves performance by 3-5% in scenario 1 and 10-15% in scenario 2. While in scenario 3, the performance of both schemes is similar.

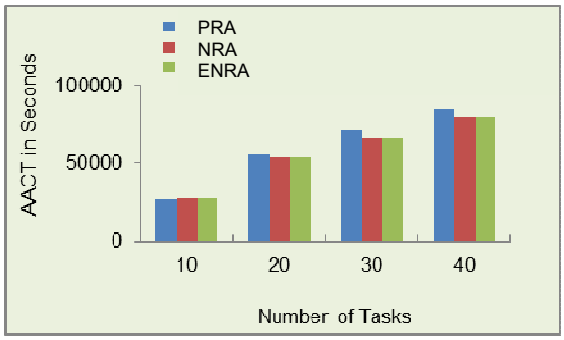

Figure 2: AACT for Scenario 1

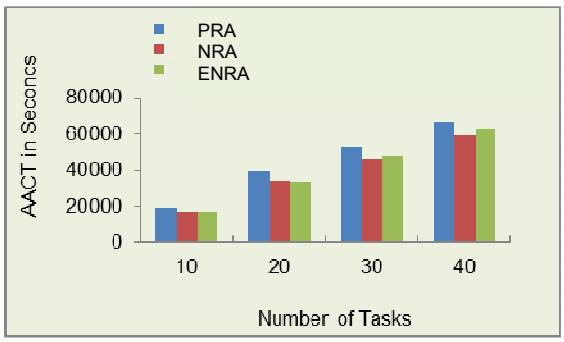

Figure 3: AACT for Scenario 2

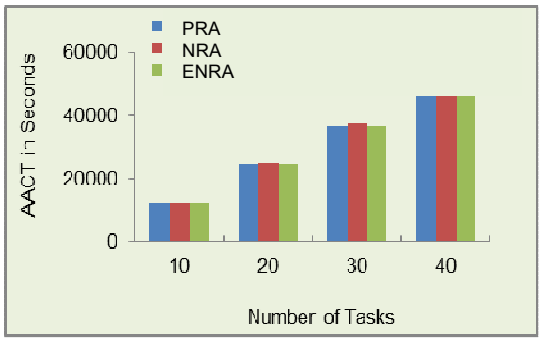

Figure 4: AACT for Scenario 3

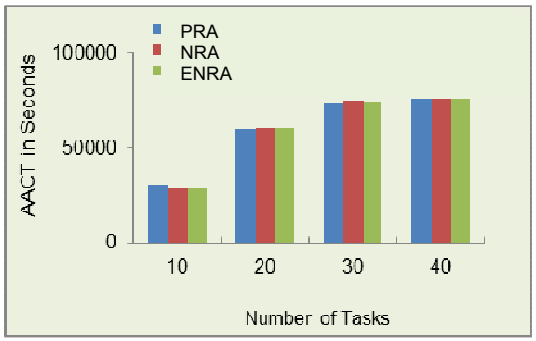

Figure 5: AACT for Scenario 4

In first scenario, 15 nodes divided into two large groups and two small groups were deployed in the region. Due to less number of nodes and group of nodes, there were few choices for allocation. Therefore, performance gains are not significant. In second scenario, large numbers of nodes were deployed into small groups, so there were several choices for allocation. NRA allocated task to nodes with better connection quality. Furthermore, when large numbers of tasks were submitted, they were evenly distributed to nodes in order to balance the data transfer load in the network. PRA allocated tasks to nodes based on transmission power and did not consider network conditions. Therefore, tasks were allocated to same group of nodes because they were accessible at minimum transmission power. This increased data transfers and degraded communication performance and connection quality between nodes in that region. Thus completion time of tasks allocated in that region was increased.

In scenario 3, both schemes have similar performance. This is because PRA did not result into network congestion or traffic overload in any region. The nodes were deployed in 5 groups of size 4. Inter-node distances in three groups were 50

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

m and in other two groups were 100 m. PRA always allocated tasks to nodes in three groups because nodes were accessible at less transmission power. The results for scenarios 4-6 are shown in Figures 5-7. NRA outperforms PRA in scenario 6 due to TCP traffic between nodes in few groups. Groups with TCP traffic were experiencing poor communication performance. NRA did not schedule tasks to that group whereas PRA allocated tasks to nodes in groups with TCP traffic.

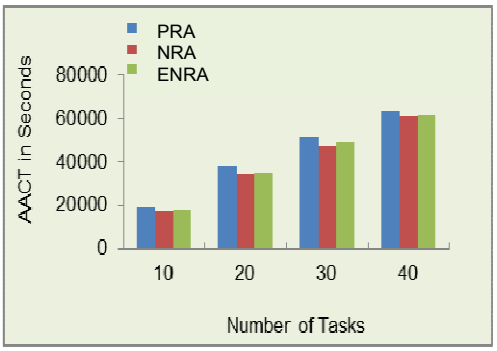

Figure 6: AACT for Scenario 5

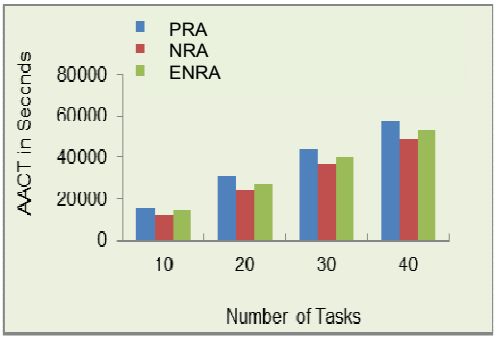

Figure 7: AACT for Scenario 6

If communication performance is similar in all network portions, the performance of ENRA and NRA is also similar. But in case of varying communication performance, ENRA achieves performance somewhere between PRA and NRA as demonstrated in Figures 3-6-7.

5.3.2 Transmission Energy Consumption

Transmission energy consumption for scenarios 1-3 is shown in Figures 8-10. In scenarios 1 and 3, PRA improves energy efficiency by 3% whereas in scenario 2, NRA performs better and reduces energy consumption by 2%. Since PRA schedules task on nodes reachable at min TPL, it improves performance in scenarios 1 and 3. In scenario 2, PRA does not perform well because it schedule task to nodes without taking into account the network information.

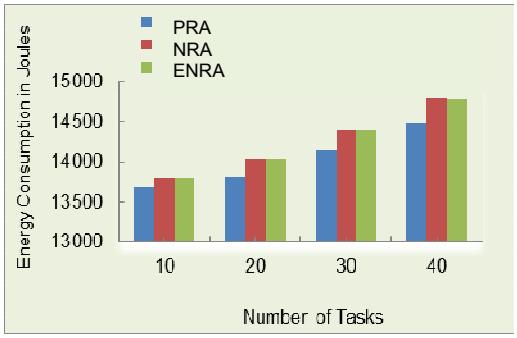

Figure 8: Energy Consumption for Scenario 1

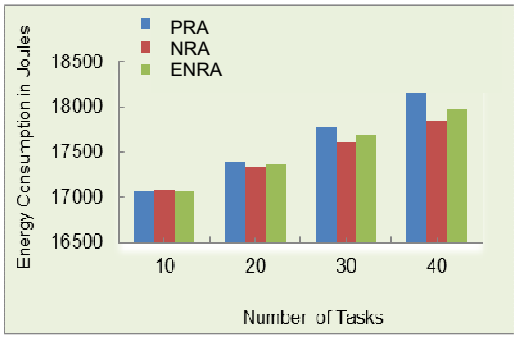

Figure 9: Energy Consumption for Scenario 2

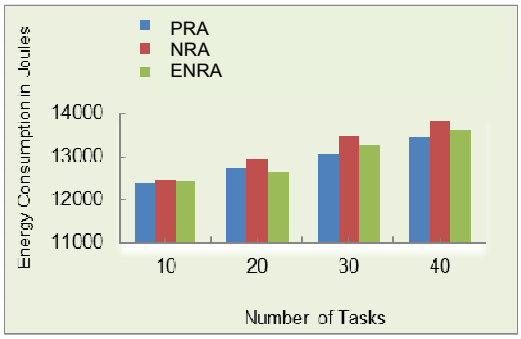

Figure 10: Energy Consumption for Scenario 3

Figures 11-13 demonstrate transmission energy consumption for scenarios 4-6. NRA performs slightly better against PRA in scenario 4 whereas in scenarios 5-6, PRA has better performance. In scenario 4, PRA increased the number of lost packets due to large amount of data transfers in some network regions and therefore consumed more energy. Theoretically, PRA should save more energy than NRA particularly in scenarios 6 due to transmission power control mechanism. But it does not achieve significant performance gains because

Journal of Theoretical and Applied Information Technology
31st January 2017. Vol.95. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

it consumes lot of energy due to increased communication cost incurred by poor resource allocation policy.

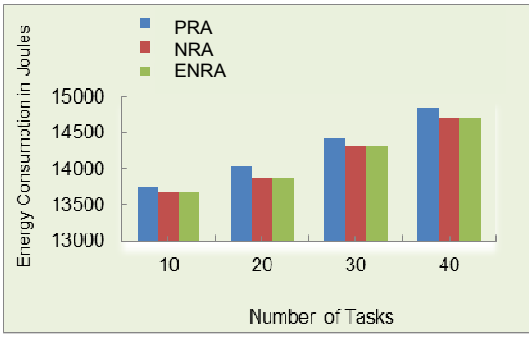

Figure 11: Energy Consumption for Scenario 4

The key factors that contribute to energy consumption are transmission power and communication cost. PRA resulted into network congestion and thus increased communication cost in both scenarios 4 and 6, but it has better performance in scenario 6 and poor in scenario 4. This is because in scenario 4 distances between nodes in group were same, so energy consumed due transmission power in both scheme was same. PRA performed worse because it consumed more energy due to increased communication cost. While in scenario 6, energy consumption due to communication cost was more than NRA, but energy saving due to transmission power control mechanism was more significant which dominated overall performance. This is because some nodes were accessible at minimum transmission power while others at maximum transmission power. PRA allocated tasks to nodes accessible to minimum transmission power and therefore conserved more energy. ENRA trades energy to reduce accumulative application completion time compared to PRA, but in scenario 5 it conserves more energy than other two schemes.

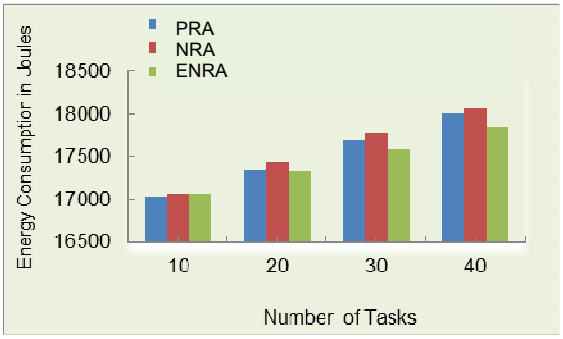

Figure 12: Energy Consumption for Scenario 5

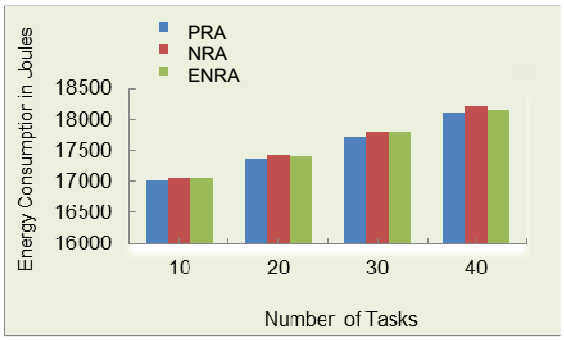

Figure 13: Energy Consumption for Scenario 6

5.3.3 Task Completion time

Task completion time for scenario 6 is given in Table 5. The amount of data transferred between tasks varied from 4-8 MB. As results show, both schemes have almost same minimum task completion time. But maximum task completion time of PRA is worse than NRA. As mentioned earlier, PRA allocates tasks to nodes where large amount of data transfers are already in progress. Allocation of additional tasks further degrades communication performance and thus increases data transfer times.

Table 5: Maximum, Minimum And Average Task Completion Time Of Scenario 6.

		PRA	NRA	ENRA
10 Tasks	**Max**	3886	1230	3858
	Min	1219	1218	1219
	Average	1517	1221	1499
20 Tasks	**Max**	4151	1235	4080
	Min	1220	1219	1218
	Average	1539	1223	1373
30 Tasks	**Max**	5949	2441	4169
	Min	1222	1220	1219
	Average	2200	1834	1988
40 Tasks	**Max**	6250	2465	6134
	Min	2453	2438	2447
	Average	2879	2447	2644

The task completion time for scenario 2 is given in Table 6. As demonstrated, both schemes have almost same minimum and maximum task completion times. But in case of average task completion time, NRA outperforms PRA. NRA has worse maximum task completion time because

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

it also allocated some tasks to nodes in overloaded network region due to unavailability of nodes. But in general very few tasks were allocated in overloaded network region compared to PRA which allocated large number of tasks and thus increased average task completion time. ENRA performs better than PRA but poor compared to NRA.

Table 6: Maximum, Minimum And Average Task Completion Time Of Scenario 2.

		PRA	NRA	ENRA
10 Tasks	Max	5300	5200	5311
	Min	1236	1234	1232
	Average	1878	1674	1674
20 Tasks	Max	5408	5335	5351
	Min	1240	1244	1219
	Average	1964	1679	1669
30 Tasks	Max	6370	6390	6333
	Min	1240	1244	1232
	Average	2640	2318	2295
40 Tasks	Max	6598	6570	6561
	Min	2482	2482	2438
	Average	3339	2960	2937

6. CONCLUSION

In this paper, an energy efficient and network aware resource scheduling scheme is proposed for scheduling of data intensive tasks on sensor cloud. The scheme uses multi-level transmission power and network information to reduce transmission energy consumption and data transfer time. The paper also investigates the relationship between data transfer time and transmission energy consumption. The performance of proposed scheme is compared with power-based resource allocation scheme and network aware resource allocation scheme. The experimental results obtained through simulations show that performance of scheme varies with respect to number of nodes, deployment mechanism, application configuration and network conditions. When some network portions are overloaded and are experiencing poor connection quality, network aware resource allocation scheme significantly reduces accumulative application completion time. Whereas with similar connection quality, the performance of all schemes with respect to accumulative application completion time is similar but in term of transmission energy consumption power-based resource allocation scheme achieves significant performance gains.

In future, we aim to extend the scheme to address the problem of scheduling real time tasks on a sensor cloud.

ACKNOWLEDGMENTS

This work was supported by Hankuk University of Foreign Studies Research Fund of 2016.

A preliminary version of this paper appeared in IEEE ICIRA 2013, September 25-28, Busan, South Korea. This version includes a detailed analysis of previous scheme and a new energy efficient and network aware resource scheduling scheme.

REFRENCES:

[1] Brian C. Mashburn; Douglas S. Blank. Graphics + Robotics + AI = Fast, 3D Scene Construction, *10th Midwest Artificial Intelligence and Cognitive Science Conference*, April 23-25, 1999, Indiana University, Bloomington, Indiana.

[2] J. Falcou; J. Serot; T. Chateau; F. Jurie. A Parallel Implementation of 3D Reconstruction Algorithm for Real-Time Vision. *Parallel Computing* 2005.

[3] Se, S.; Lowe, D.G.; Little, J.J. Vision-based global localization and mapping for mobile robots *Robotics, IEEE Transactions on*, vol.21, no.3, pp. 364- 375.

[4] Sayed Chhattan Shah; Myong-Soon Park. An Energy-Efficient Resource Allocation Scheme for Mobile Ad Hoc Computational Grids, *Journal of Grid Computing Journal*, Apr.16,2011, no.10723.

[5] Sayed Chhattan Shah et al. Network Aware Resource Allocation Scheme for Mobile Ad Hoc Computational Grid, *Intelligent Robotics & Applications, Lecture Notes in Computer Science*, 2013, Vol. 8102, pp.105-116.

[6] Shah, Sayed Chhattan. Energy efficient and robust allocation of interdependent tasks on mobile ad hoc computational grid, Concurrency *Computat.: Pract. Exper.,* 10.1002/cpe.3297, 2014.

[7] S C Shah et al. An effective and robust two-phase resource allocation scheme for interdependent tasks in mobile ad hoc

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

computational Grids, *Journal of Parallel and Distributed Computing*, 2012.

[8] V. Kawadia; P. R. Kumar. Principles and Protocols for Power Control in Ad Hoc Networks, *IEEE J. Select. Areas Commun*. 2005, 15–16.

[9] B. G. Chun; P. Maniatis. Augmented Smartphone Applications Through Clone Cloud Execution, *12th Workshop on Hot Topics in Operating Systems (HotOS XII)*. Monte Verita, Switzerland: USENIX, 2009.

[10] S. Garriss; R. Caceres; S. Berger; R. Sailer; L. van Doorn; ´X. Zhang. Trustworthy and Personalized Computing on Public Kiosks, 6th *International Conference on Mobile Systems, Applications, and Services (MobiSys '08),* 2008, pp. 199 – 210.

[11] M. Satyanarayanan; P. Bahl; R. Caceres; N. Davies. The Case for VM-Based Cloudlets in Mobile Computing, *IEEE Pervasive Computing*, 2009, 8 (4), pp. 14–23.

[12] J. Rellermeyer; O. Riva; G. Alonso. AlfredO: Architecture for Flexible Interaction with Electronic Devices, *9th ACM/IFIP/USENIX International Conference on Middleware (Middleware 2008),* Lecture Notes in Computer Science, Vol. 5346. Leuven, Belgium: Springer, 2008, pp. 22–41.

[13] J. S. Rellermeyer; G. Alonso; T. Roscoe. R-OSGi: Distributed Applications through Software Modularization, ACM/IFIP/USENIX *8th International Middleware Conference (Middleware 2007).* Newport Beach, CA, USA: Springer, Nov. 2007, pp. 50–54.

[14] R. K. Balan; M. Satyanarayanan; S. Y. Park; T. Okoshi. Tactics-based Remote Execution for Mobile Computing, *Int. Conf. Mobile Systems, Applications and Services*, San Francisco, May 5–8, 2003.

[15] B.-G. Chun; S. Ihm; P. Maniatis; M. Naik; A. Patti. Clonecloud: elastic execution between mobile device and cloud. *In ACM EuroSys, 2011.*

[16] E. Cuervo; A. Balasubramanian; D. K. Cho; A. Wolman; S. Saroiu; R. Chandra; P. Bahl. Maui: making smartphone last longer with code offload. *In ACM MobiSys, 2010.*

[17] M.A.M. Mohamed et al. MOSET: An anonymous remote mobile cluster computing paradigm, *J. Parallel Distrib. Comput.* 65 (2005) 1212 – 1222.

[18] Z.-L. Zong; M. Nijim; X. Qin. Energy-Efficient Scheduling for Parallel Applications on Mobile Clusters, Cluster Computing: *Journal of Networks, Software Tools and Applications*, 2008, 11 (1), pp. 91 - 113.

[19] Chunlin Li; Layuan Li. Tradeoffs between energy consumption and QoS in mobile grid, *Journal of Supercomputing*, 2011, 55 (3).

[20] K. A. Hummel; G. Jelleschitz. Robust De-centralized Job Scheduling Approach for Mobile Peers in Ad Hoc Grids," *7th IEEE Int. Symp. Cluster Computing and the Grid*, May 14–17, 2007.

[21] D. C. Chu; M. Humphrey. Mobile OGSI.NET: Grid Computing on Mobile Devices, 5th *IEEE/ACM Int. Workshop Grid Computing*, Nov. 8, 2004.

[22] C. Li; L. Li. Utility-based Scheduling for Grid Computing under Constraints of Energy Budget and Deadline, Comput. *Stand. Interfaces,* 2009, 31 (6), pp. 12–40.

[23] Chunlin Li; Layuan Li. Tradeoffs between energy consumption and QoS in mobile grid, *Journal of Supercomputing, 2011, 55 (3).*

[24] Chunlin Li; Layuan Li. Collaboration among mobile agents for efficient energy allocation in mobile grid, *Information Systems Frontiers,* 2012, 14 (3), pp. 711-723.

[25] H. Liu; T. Roeder; K. Walsh; R. Barr; E. G. Sirer. Design and Implementation of a Single System Image Operating System for Ad Hoc Networks, *3rd Int. Conf. Mobile Systems, Applications, and Services*, June 6–8, 2005, pp. 149–162.

[26] Antonio Tadeu A. Gomes et al. DICHOTOMY: A Resource Discovery and Scheduling Protocol for Multihop Ad Hoc Mobile Grids, *7th IEEE Int. Symp. Cluster Computing and the Grid,* May 2007, pp. 14–17.

[27] V. Vetri Selvi; S. Sharfraz; R. Parthasarathi. Mobile Ad Hoc Grid Using Trace Based Mobility Model, *GPC 2007, LNCS 4459*, 2007, pp. 274–285.

[28] S. Shilve; H.J. Siegel; A.A. Maciejewski; P. Sugavanam; T. Banka; R. Castain; K. Chindam; S. Dussinger; P. Pichumani; P. Satyasekaran; W. Saylor; D. Sendek; J. Sousa; J. Sridharan; J. Velazco. Static Allocation of Resources to Communicating Subtasks in a Heterogeneous Ad Hoc Grid Environment, *J. Parallel Distrib. Comput,* 2006, 66 (4), pp. 600–611.

[29] Cong Shi; Vasileios Lakafosis; Mostafa H. Ammar; Ellen W. Zegura. Serendipity: enabling remote computing among intermittently connected mobile devices, *Proceedings of the*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing, 2012.

[30] J.M. Rodriguez; C. Mateos; A. Zunino. Energy-efficient Job Stealing for CPU-intensive processing in Mobile Devices, *Computing*, Springer, In Press, 2012.

[31] M. Yuriyama; T. Kushida. Sensor-cloud infrastructure: physical- sensor management with virtualized sensors on cloud computing, *in Network-Based Information Systems, 13th IEEE International Conference on.* 2010, pp. 1-8.

[32] H. Viswanathan; E. K. Lee; D. Pompili. Mobile grid computing for data- and patient-centric ubiquitous healthcare, in *Proc. of IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things*, Seoul, Korea, June 2012.

[33] H. Kim; Y. el Khamra; I. Rodero; S. Jha; M. Parashar. Autonomic Management of Application Workflows on Hybrid Computing Infrastructure, *Telecomm. Sys*., Feb. 2011, vol. 19, no. 2-3, pp. 75–89.

[34] C. K. Tham; R. Buyya. SensorGrid: Integrating Sensor Networks and Grid Computing. *CSI Communications,* July 2005, 29(1), pp. 24-29.

[35] M. Gaynor; S.L. Moulton; M. Welsh; E. LaCombe; A. Rowan; J. Wynne. Integrating wireless sensor networks with the grid, *IEEE Internet Computing,* 2004, 8 (4), pp. 32–39.

[36] H.B. Lim; P. Mukherjee1; V.T Lam; W.F. Wong; S. See. Sensor Grid: Integration of Wireless Sensor Networks and the Grid, Proceedings of *30th IEEE Conference on Local Computer Networks,* November 2005, pp. 91-98, Sydney, Australia.

[37] T. Vigneswari; M. A. Maluk Mohamed. Scheduling in Sensor Grid Middleware for Telemedicine Using ABC Algorithm, *International Journal of Telemedicine and Applications,* 2014, Vol. 2014, 7 pages.

[38] A. Alamri et al. A survey on sensor-cloud: Architecture, applications, approaches, *Int. J. Distributed Sensor Networks,* V 2013, pp. 1–18.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Sort tasks according to precedence and parallel execution dependencies
While (tasks are not allocated) **do**
Get a task t_i or tasks set T^* from application dependency graph
If (t_i is an independent task) **then**
If (t_i is computation-bound task $t_i^{cpu-bound}$) **then** allocate to high processing node
Else (t_i is local or remote communication-bound task) allocate to low processing node
Else If (interdependent tasks set T^*) **then** Select a node with highest grade max (n_i^g) within sensor cloud
allocateTasks (T^*, max (n_i^g))
Else /* one task t_r is already allocated and dependent tasks T^* need allocation */
If (more than one dependent tasks - count (T^*) > 1) **then**
Get a node n_r that hosts an already allocated task t_r
If (t_r or successor(t_r) is rc-bound and (T^* - successor(t_r)) are not rc-bound) **then**
Select a node with highest grade max(n_i^g) within a range of n_r and allocateTasks(T^*, max(n_i^g))
Else Select a node with highest grade max(n_i^g) within ad hoc sensor cloud
allocateTasks(T^*, max(n_i^g))
Else /*only one dependent task*/
Get a node n_r that hosts already allocated task t_r
Get neighbor nodes of n_r
If (neighbor nodes >1) **then**
If (t_i is computation-bound task $t_i^{cpu-bound}$) **then** allocate to high processing neighbor node
Else If (t_i is local or remote communication-bound task) **then** allocate to low processing neighbor node
allocateTasks(T^*, n_i **)** {
For each task within tasks set T^* {
If (t_i is computation-bound task $t_i^{cpu-bound}$) **then**
allocate to high processing node $n_j \in n_i^{kNN}$
Else (t_i is local or remote communication-bound task) **then**
allocate to low processing node $n_j \in n_i^{kNN}$
}

Algorithm 1: Pseudo Code Of Network Aware Resource Scheduling Scheme For Sensor Cloud