# Effect of RPGM and Gauss Markov Node Mobility for Resource Allocation Performance in Mobile Ad Hoc Computational Grid

**Sri Chusri Haryanti and Riri Fitri Sari**,

Departement of Electrical Engineering, Faculty of Engineering, Universitas Indonesia,
Kampus Baru UI Depok 16424, Indonesia

**Summary**
Mobile ad hoc computational grid enables resource sharing between mobile nodes in an ad hoc networks for completing specific tasks. To maximize utilization of sharing resources in such grid, an effective resource allocation algorithm that is suitable with the environment, plays a key role. In this paper, we investigate the performance of resource allocation scheme EERA that has been designed for mobile ad hoc computational grid in RPGM and Gauss-Markov node mobility model. We perform the simulation in NS2. The result shows that accumulative application completion times for the scheme using Gauss-Markov mobility nodes is 45% up to 350% greater than using RPGM model. Increasing the maximum node's speed will increase the accumulative application completion times. Meanwhile the average end-to-end delay and the number of drop packet for Gauss-Markov is lower compare to RPGM in maximum speed of 5m/s.

***Key words:***
*resource allocation, mobile ad hoc computational grid, node mobility.*

## 1. Introduction

In recent years, mobile ad hoc computational grid is one of challenging research topics. Mobile ad hoc computational grid allows collaboration of mobile nodes to build a Grid, join existing Grid or contribute in a Grid spontaneously. Nodes can be dynamically request and receive a sharing of resources in Grid computing, and be active in the services offered by other nodes in the Grid. Mobile Ad hoc computational grid facilitate autonomous interaction without requiring pre-configuration or policy management. [1]

In order to implement mobile ad hoc grid, there are many studies to be done. Both in terms of mobile ad hoc networks, such as routing protocols, and from the side of grid computing such as resource discovery, resource allocation, job scheduling and application.

Resource allocation plays an important role in determining performance of the overall grid system. Resource allocation algorithm that is suitable for mobile ad hoc environment is expected to minimize task completion time. The challenge in designing the resource allocation algorithm for mobile ad hoc computational grid is associated with lack of infrastructure in the network, node mobility and limited power of mobile nodes. It is necessary to consider nodes mobility model to obtain good system performance. This is due to the factor that in mobile ad hoc computational grid, a node can join and leave the grid dynamically associated with mobility.

In this paper, we observe the effect of two nodes mobility models, i.e RPGM and Gauss Markov in resource allocation service in mobile ad hoc computational grid. Energy Efficient Resource Allocation (EERA) [2] is the sample of resource allocation algorithm for our investigation. EERA is a resource allocation algorithm that is designed for military application. Besides providing energy-efficient environments, EERA proved to give application completion time and the average end-to end delay that is shorter than distance-based resource allocation (DRA) [3] scheme for group mobility model [2].

This paper is organized as follow. In section II, we overview the Mobile Ad Hoc Computational Grid, Resource Allocation, Energy-efficient Resource Allocation and Mobility Model. Section III describes the simulation scenario, parameters and simulation result and analysis. Finally, we draw some conclusion and propose future work in Section IV.

## 2. Background

### 2.1 Mobile Ad Hoc Computational Grid

Grid computing is a system that enable resource sharing and problem solving between connected nodes. Based on connect and share approach, it provides high performance and cost-effective solutions. There are various service that can be provided by a grid, from data, information and knowledge services to application, storage and computational services. A computational grid is usually used to solve computationally intensive problems [4].

Computational Grids are among the first type of Grid systems. They were developed due to the need to solve problems that require processing a large quantity of operations or data. One of the main objectives of the

Manuscript received June 5, 2011
Manuscript revised June 20, 2011

Computational Grid is to benefit from the existence of many computational resources through the sharing [5].
Mobile ad hoc computational grid is the integration of computational grid and mobile ad hoc network that allow autonomous mobile devices to form a single, unified computing resource without support of any fixed infrastructure [6].
Mobile ad hoc grid allows nodes to spontaneously form an ad hoc network, forming a grid or join an existing grid, dynamically contribute to the Grid, and being active in the services offered by other nodes in the Grid. Mobile ad hoc grid facilitate autonomous interaction without requiring pre-configuration or policy management.
To maximize the utilization of shared computational resources in mobile ad hoc grid, an effective resource allocation algorithm that is suitable with the environment, plays a key role. Resource allocation is concerned with allocating the best chosen computational resources for requiring tasks.

## 2.2 Resource Allocation

Resource allocation in grid computing system is the second phase of job scheduling process. J.M. Schopf [7], define the stages in job scheduling to include resource discovery, resource allocation (system selection) and job execution. In resource allocation phase, the system will select a single resource (or one set of resources) from a group of possible resources which meet the minimum requirements for the task. The selection is generally executed through two steps, that is gathering information of resources in the system and making a decision. Allocating task in appropriate resources can minimize the computation time and increase the system performance.
Resource allocation scheme for mobile ad hoc grid is very challenging. It has to find the best resource to fulfill task requirement in heterogeneous environment. Moreover, resource allocation algorithm should also concern the dynamic of the network topology cause by the mobility of nodes.
Presently only a very few schemes of resource allocation in mobile ad hoc computational grids have been proposed. Some of them have addressed the issues such as node mobility, energy management, and task failure. A distributed resource allocation scheme based on first come first serve strategy has been proposed by Hummel and Jelleschitz [8]. Each mobile node is allowed to perform a mapping based on job's requirements. It supports tasks redundant execution and employs both proactive and reactive fault tolerance mechanism to address task failure.
To select most suitable node for task execution, Gomes et. al in [9] proposed a scheme which utilizes a delayed reply mechanism. It also provides load balancing and scalability. Node mobility has been addressed in [10] by profiling regular movements of a user over the time. The profiling records user's visited locations and associated time duration at those locations. A node which stayed longer at the location is selected for task execution. The schemes which are based on a decentralized architecture as have been mentioned before, results in poor allocation decisions due to lack of network-wide view. The tasks types and dependencies among them are not considered. The schemes are targeted towards the load balancing, scalability, and fault tolerance rather than application performance. Moreover, very few schemes have addressed node mobility which is also based on a reactive approach that degrades the performance of an application in terms of a delay to make a decision. To deal with the precedence dependencies Shilve et.al in [11] have proposed a scheme based on a static allocation of resources in ad hoc computational grids; however, due to static allocation, this scheme is not adaptive to network changes and application behavior.
We choose resource allocation scheme energy-efficient resource allocation (EERA) in [2] to be evaluated in different nodes mobility model due to the fact that EERA do not rely on any particular architecture and it address task dependencies.

## 2.3 Energy-efficient Resource Allocation (EERA)

One of very few resource allocation algorithm that have been proposed for mobile ad hoc grid is energy-efficient resource allocation (EERA). One of EERA's advantage is efficient in energy consumption. That is because EERA classify the tasks based on task type and type of task dependency (see Fig 1) and then allocate them to the nodes that can be reached with minimum power.

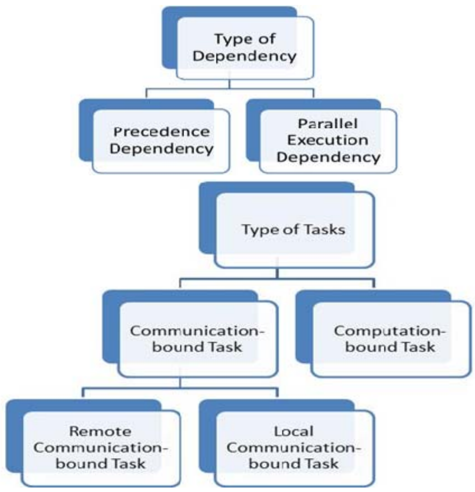


Fig. 1. Classification of task in EERA.

The task is define as an indivisible work units. The dependencies are divided into two categories: precedence dependencies and parallel execution dependencies. The tasks with precedence dependencies execute independently but require inputs generated by predecessor tasks while tasks with parallel execution dependencies periodically exchange data one another and communication among tasks may take place anytime during execution.
In addition to dependencies, tasks are also divided into three categories: computation-bound tasks, local communication-bound tasks, and remote communication-bound tasks. computation-bound tasks exchange small quantity of data and have high processor utilization while communication-bound tasks exchange large quantity of data and have low processor utilization. Among communication-bound tasks, local communication-bound tasks spend most of the time performing local I/O operations while remote communication-bound tasks spend most of the time performing remote I/O operations. The local I/O operations involve data transfer from/to local storage while remote I/O operations involve data transfers across the network.
The purpose of dependencies and tasks classification is to exploit them in order to improve utilization of computing resources and application performance in resource allocation scheme. Every node in EERA runs k-nearest neighbor search algorithm to find nodes that are accessible in one hop in different transmission power levels. It is shown in [2] that EERA gives lower accumulative application computation time and average end-to end delay compare to distance-based resource allocation scheme..
EERA is assumed that mobile nodes move in a group and it uses group mobility model. However, mobile ad hoc computational grid is also expected to be used for other applications, so it should be robust to nodes mobility model.
In this research, we investigated EERA's performance on its implementationimplementing in different node mobility model, i.e reference point group mobility model (RPGM) and Gauss Markov (GM) model. This investigation is expected to be a baseline in developing a resource allocation scheme which can be used in a more general application in mobile ad hoc computational grid.

## 2.4 Mobility Model

The mobility model is designed to describe the movement pattern of mobile user, and how their location, velocity and acceleration change over time [12]. It is important to include mobility pattern of users because it plays a significant role in determining system performance.

There are many mobility model that can be used. In this paper, we only described several models that is widely used. A complete survey of the mobility models for ad hoc networks can be found in [12].

### 2.4.1 Random Waypoint Model

In this model, as described by its name, the users move randomly. The mobile nodes move randomly and freely without restrictions. The destination, speed and direction are all chosen randomly and independently from other nodes. Because of its simplicity and wide availability, this model has become a "benchmark" to evaluate the mobile ad hoc network routing protocols.

### 2.4.2 Random Walk Mobility Model

This model is a variant of Random Waypoint Model. A mobile node moves from its current location to a new location by randomly choosing a direction and speed in which to travel. The random direction and the random speed are chosen from pre-defined ranges $[0,2\pi]$ and $[V_{min}, V_{max}]$, respectively. Each movement in the Random Walk Mobility Model occurs in either a constant time interval t or a constant distance traveled d, at the end of which a new direction and speed are calculated. If a mobile node which moves according to this model reaches a simulation boundary, it "bounces" off the simulation border with an angle determined by the incoming direction. The mobile node then continues along this new path

### 2.4.3 Random Point Group Mobility

Group mobility is usually used in military battlefield communication. In RPGM, the nodes are divided into groups. Each group of nodes has a group leader that determines the group's motion behavior. Initially, each member of the group is uniformly distributed in the neighborhood of the group leader. Every node has a speed and direction that is derived by randomly deviating slightly from that of the group leader. The speed deviation is set according to the speed deviation ratio (SDR) and the angle deviation ratio is set according to the angle deviation ratio (ADR) as follows:

$$|V_{node}(t)| = |V_{reference}(t)| + random()xSDRxV_{max} \quad (1)$$

$$\theta_{node}(t) = \theta_{reference}(t) + random()xADRx\theta_{max} \quad (2)$$

In the above expressions, random() refers to a uniformly distributed random number between [0,1]. RPGM provides high spatial correlation between nodes, which leads to high link durations and less change in the relative network topology.

#### 2.4.4 Gauss Markov Mobility Model

In this model, the velocity of mobile node is assumed to be correlated over time and is modeled as a Gauss-Markov stochastic process. It is a temporally dependent mobility model where the degree of dependency is determined by the memory level parameter α. By tuning this parameter various scenarios are obtained: (i) $\alpha = 0$ then the model is memoryless, (ii) $\alpha = 1$ then the model has strong memory and (iii) $0 < \alpha < 1$ then the model has some memory.

In this research we only use RPGM to prove EERA's performance in group mobility model and Gauss Markov to gain information on the performance of resource allocation scheme in a more random nodes mobility.

## 3. Simulation

### 3.1 Simulation Scenario

The simulation have been carried out using the Network Simulator (NS) version 2.34. We assume that the transmission range, the maximum possible distance between two communication mobile nodes is 250 m.
In our scenario, the ratio of remote communication-bound to local communication and computation-bound task is 2:3, and ratio of tasks with parralel execution dependencies to task with precedence dependencies is 3:2
Implementation of EERA is investigated in RPGM and Gauss-Markov mobility model that is generated using BonnMotion tools [13]. For each model we varied the maximum speed of mobile nodes (s) for s = 1m/s and s = 5 m/s.

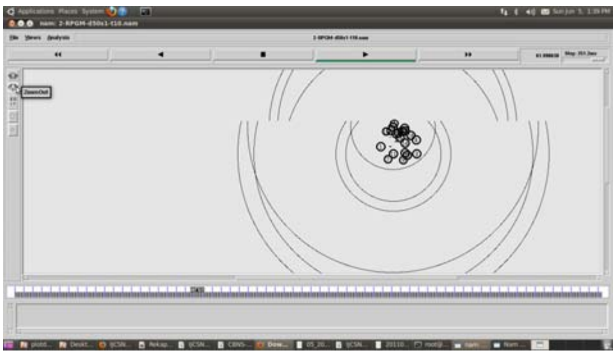

Fig. 2. Simulation Scenario.

### 3.2 Parameters

The parameters used in the simulation are given in Table 1.

Table 1: Simulation Parameters

| Parameter | Value |
|---|---|
| Range transmission | 250 m |
| Simulation time | 100 s |
| Topology | 600 x 600 m |
| Nodes number | 20 |
| Transmission Power Level | 6 |
| Number of task | 10-20-30-40 |
| Transport Protocol | TCP |
| Routing Protocol | Clusterpow |
| MAC Protocol | IEEE 802.11 |
| Packet rate | 5 packet/s |
| Packet size | 512 bytes |
| Node mobility Model | RPGM & GM |
| Maximum speed | 1 m/s and 5 m/s |

Some parameters evaluated and analyzed based on the simulation are :
1) accumulative application completion time
The task completion time consists of two key components: execution cost and communication cost. The communication cost is a product of average communication delay and the number of packets transmitted between tasks. The equation to calculate accumulative task completion time is decribe as in equation (3) – (5) as follow :

$$AACT = \sum T_{CompTime} \quad (3)$$
$$T_{CompTime} = E_C + C_C \quad (4)$$
$$C_C = P_n \times D_{end-end} \quad (5)$$

Where :
TcompTime = Task completion time
$E_C$ = Execution cost
$C_C$ = Communication cost
$P_n$ = Number of packets
$D_{end-end}$ = Average end-to-end delay

2) average end-to-end delay
The average end-to-end communication delay refers to the time taken for a packet to be transmitted across the network from a source to destination.

### 3.3 Simulation Result and Analysis

#### 3.3.1 Accumulative Application Completion Time

Fig. 3, 4, 5 and 6 demonstrate the effect of mobility model in resource allocation algorithm for accumulative application completion time.

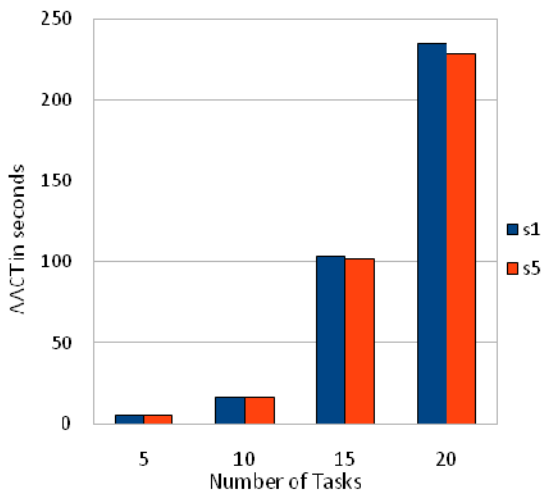


Fig.3. Accumulative Application Completion Time of EERA using RPGM mobility model.

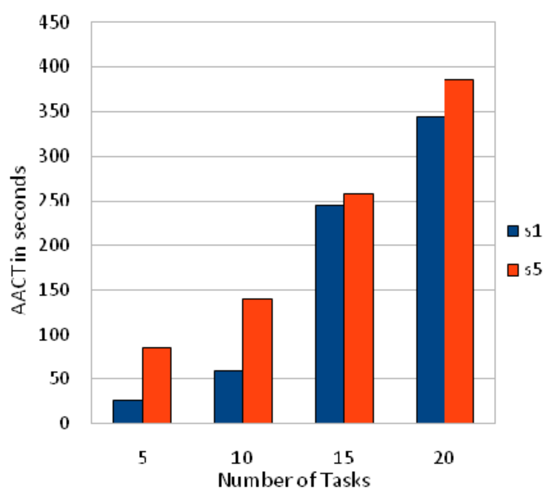


Fig.4. Accumulative Application Completion Time EERA using GM mobility model.

It is shown in Fig. 3 that for RPGM, at maximum speed of s=5 m/s, the accumulative application completion time is lower than at in s=1 m/s. As mentioned previously, two key components which contribute to the task completion time are the execution cost and communication cost. Due to the fact that communication cost is the product of the average end-to-end delay and number of packet transmitted among tasks, greater speed will decrease the range of mobile nodes time-by-time so it decrease propagation delay and also application completion time.

For Gauss-Markov mobility model, in contrast, greater speed will cause higher accumulative application completion time (Fig. 4). This is because in Gauss-Markov mobility model, greater speed will fastly increase the distance between mobile nodes. Automatically, this will increase the propagation delays and also the task completion times.

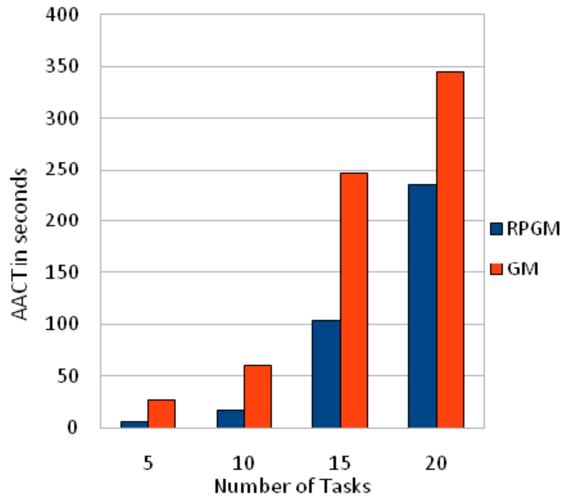


Fig.5. Accumulative Application Completion Time of EERA using RPGM vs GM for s=1m/s

The comparison of RPGM and Gauss-Markov mobility model for different node with maximum speed are depicted in Fig 4 and Fig. 5. It is shown that for both s=1 m/s and s=5 m/s, accumulative application completion time of EERA in GM exceed RPGM. For s=1 m/s, the EERA's AACT in GM is 45% to 350% higher than in RPGM and for s=5 m/s is 70% to 140% higher than in RPGM. This proved that EERA should be modified for application with more random users mobility pattern.

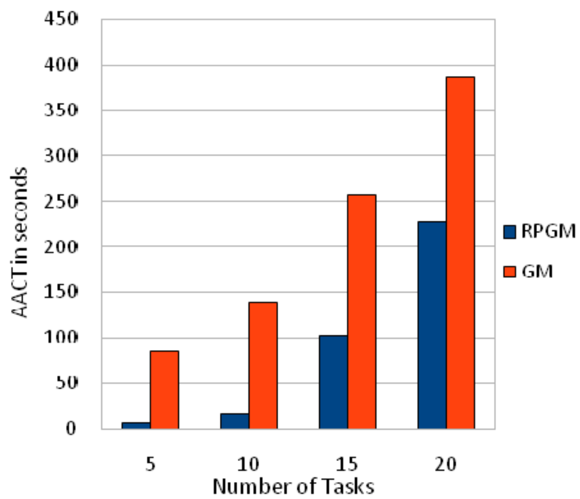


Fig.6. Accumulative Application Completion Time of EERA using RPGM vs GM for s=5m/s

### 3.3.2 Average End-to-end Delay

Fig. 7, 8, 9 and 10 demonstrate the average end-to-end delay in resource allocation algorithm for RPGM and Gauss Markov mobility model.

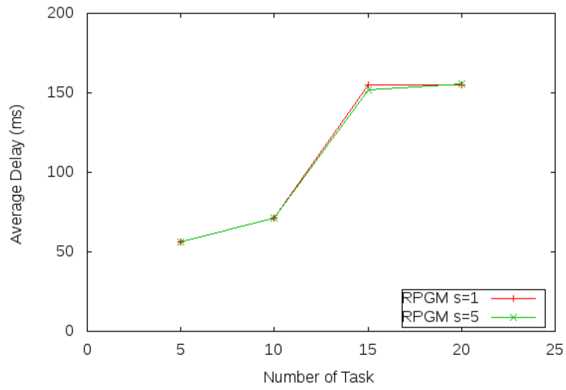


Fig.7. Average End-to-end delay of EERA using RPGM mobility.

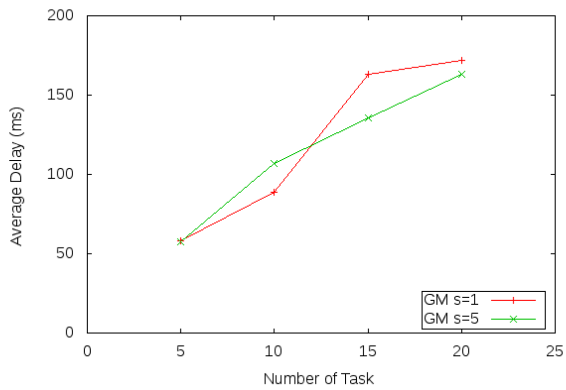


Fig.8. Average End-to-end delay of EERA using Gauss Markov Mobility.

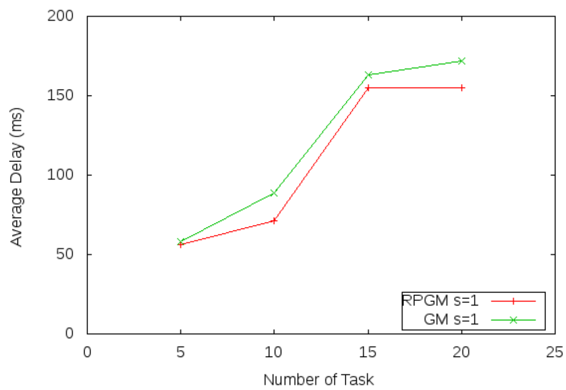


Fig.9. Average End-to-end delay of EERA using RPGM vs GM mobility model for s=1m/s

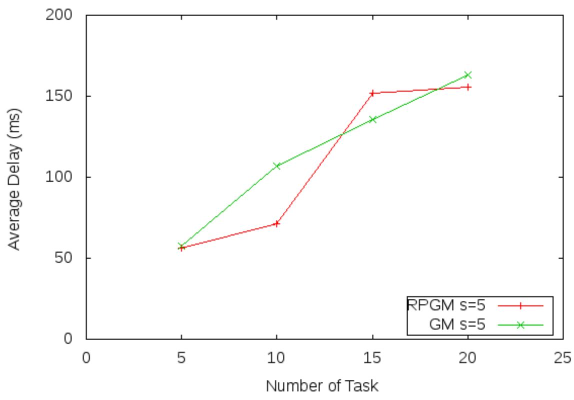


Fig.10. Average End-to-end delay of EERA using RPGM vs GM mobility model for s=5m/s

It is shown in Fig. 7 that RPGM results in just slightly different average end-to-end delay for different maximum mobile node speed. For Gauss-Markov mobility model, in Fig. 8, the average end-to-end delay for s=5 m/s is higer compared with s=1 m/s.

Fig. 9 and 10 show the comparison of EERA's average end-to-end delay in RPGM and Gauss-Markov mobility model. It is shown that for s=1 m/s the average end-to-end delay RPGM is lower than Gauss-Markov (approximately 10%), but for s=5 m/s, average end-to-end delay of RPGM is greater than Gauss-Markov (about 5%). This can be explained that the distance between nodes in RPGM is relatively "constant" to each other time-to-time. Meanwhile greater speed in Gauss-Markov will increase the probability of nodes reaches the minimum distance among them.

### 3.3.3 Drop Packet

The effect of RPGM and GM mobility model in EERA for the number of drop packet are shown in Fig. 11, 12, 13 and 14.

The effect of both mobility model for EERA in the number of drop packet is similar to the effect of the average end-to-end delay.

It is shown in Fig. 11 that RPGM results in almost the same number of drop packet on different maximum mobile node speeds. For Gauss-Markov mobility model, Fig. 12 shows that the number of drop packet for s=5 m/s is less compared to s=1 m/s. Fig. 13 and 14 show the comparison of EERA's number of drop packet in RPGM and Gauss-Markov mobility model. It is shown that for s=1 m/s the number of drop packet RPGM is lower than Gauss-Markov (approximately 10%), but for s=5 m/s, the number of drop packet RPGM is greater than Gauss-

Markov (approximately 8%). This is due to the same reason the average end-to end delay case.

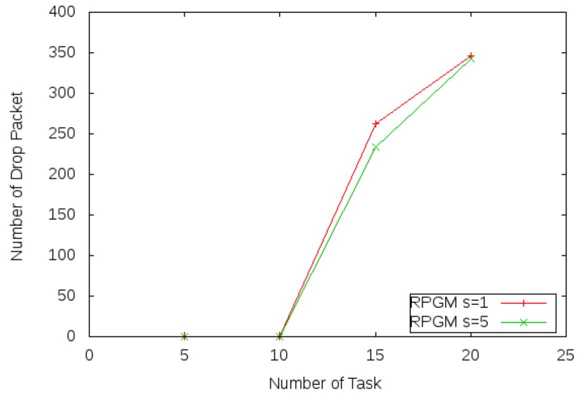


Fig.11. Number of Drop Packet of EERA using RPGM mobility.

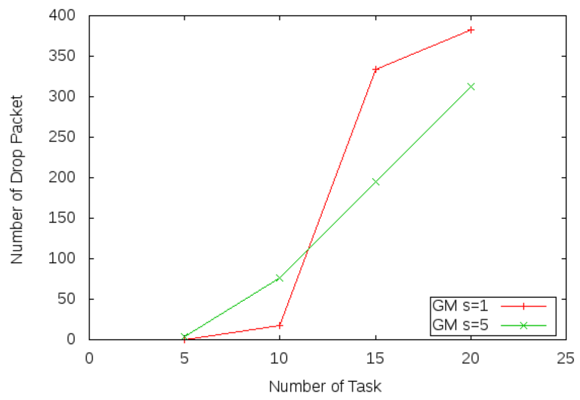


Fig.12. Number of Drop Packet of EERA using Gauss Markov Mobility.

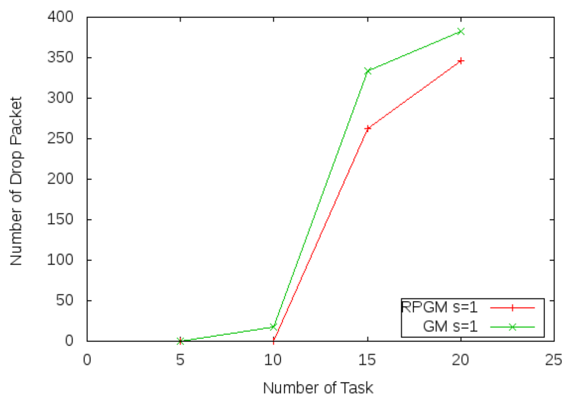


Fig.13. Number of Drop Packet of EERA using RPGM vs GM mobility for s=1m/s

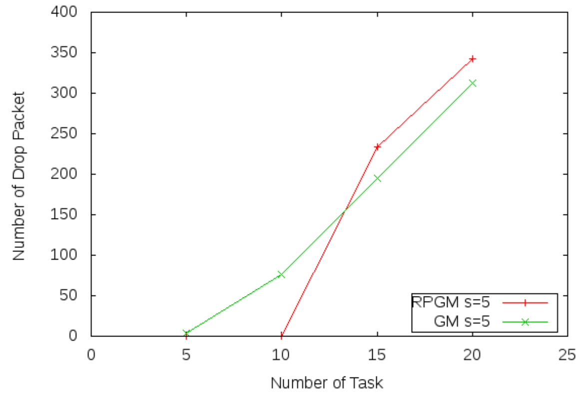


Fig.14. Number of Drop Packet of EERA using RPGM vs GM mobility for s=5m/s

## 3. Conclusion and Future Works

The simulation result shows that different node mobility pattern will give different performance for resource allocation scheme. For more random mobility model such as Gauss-Markov model, the accumulative application completion time of resource allocation scheme is 45% up to 350% greater compare to RPGM. Increasing the maximum speed of mobile node will also increase the accumulative application completion time because increasing speed will increase the rate of increasing distance of nodes. Gauss-Markov mobility model provides higher average end-to-end delay for s=1 m/s compare to RPGM, but lower for s=5 m/s.

In order to be able to be used in a more general environment, a resource allocation scheme has to give better performance in random nodes mobility. In the future, we plan to modify EERA's algorithm in order to be able to address node mobility pattern.

### Acknowledgments

We would like to thank Mr. Sayed Chhattan Shah from Korea University for disscusion and sharing.

## References

[1] K. Amin, G. Laszewski, and A. Mikler, "Toward an Architecture for Ad Hoc Grids," Proc. 12th Int'l on. Advanced Computing and Communications (ADCOM 04), 2004, www-unix.mcs.anl.gov/~laszewsk/papers/vonLaszewski-adhoc-adcom2004.pdf.

[2] S.C Shah, S.H. M.S.Park, "An Energy-Efficient Resource Allocation Scheme for Mobile Ad Hoc Computational Grid", Journal of Grid Computing, 2011

[3] S.C Shah, S.H. Chauhdary, A.K. Bashir, M.S.Park, "A Centralized Location-Based Job Scheduling Algorithm for Inter-dependent Jobs in Mobile Ad Hoc Computational Grids", Journal of Applied Sciences 10 (3); 174-181, 2010
[4] M. Baker, R. Buyya and L. Domenico, Grids and Grid Technologies for Wide Area Distributed Computing, Software Practice and Experience. John Wiley and Sons, Ltd, UK, 2002
[5] F. Xhafa and A. Abraham. Computational models and heuristic methods for grid scheduling problems. Future Generation Computer Systems, Elsevier, 26(4):608–621, 2010.
[6] D.C. Marrinescu, G. Marinescu, Y. Ji, L. Boloni and H.J. Siegel, Ad hoc Grids: Communication and Computing in power Constrained Environment, Proceedings of Workshop on Energy-Efficient Wireless Communications and Networks, April 9 -11, Phoenix, Arizona, pp:113-122, 2003
[7] J. Schopf. Ten Actions when Grid Scheduling. In Grid Resource Management (J. Nabrzyski, J. Schopf and J. Weglarz, eds.), pages 15-23, Kluwer Academic Publishers, 2004.
[8] K A Hummel and G Jelleschitz, "Robust de-centralized job scheduling approach for mobile peers in ad hoc grids" Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07), IEEE Computer Society, 2007
[9] A.T.A. Gomes et.al, “DICHOTOMY: A Resource Discovery and Scheduling Protocol for Multihop Ad hoc Mobile Grids” Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid'07), 2007.
[10] V. Vetri Selvi, Shakir Sharfraz, Ranjani Parthasarathi, “Mobile ad hoc grid using trace based mobility model”, GPC 2007, LNCS 4459, pp. 274–285, 2007.
[11] Shilve, S., Siegel, H.J., Maciejewski, A.A., Sugavanam, P., Banka, T., Castain, R., Chindam, K., Dussinger, S., Pichumani, P., Satyasekaran, P., Saylor, W., Sendek, D., Sousa, J., Sridharan, J., Velazco, J.: Static allocation of resources to communicating subtasks in a heterogeneous ad hoc Grid environment. J. Parallel Distrib. Comput. 66(4), 600–611, 2006
[12] F. Bai and A. Helmy, “A Survey of Mobility Models,” book chapter in Wireless Ad-Hoc Networks, Springer, October 2006. http://nile.usc.edu/important/chapter1.pdf
[13] BonnMotion, a Mobility Scenario Generation and Analysis Tool, Version December 22, 2010, http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion/