View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Revistas Científicas Indexadas Universidad Surcolombiana

Performance Comparison of Scheduling Techniques to Manage Transactions for Real-Time Mobile Databases in Ad Hoc Networks

Le Gruenwald[1], Matilde Montealegre[2], Chuo N. Lau[1]
[1]University of Oklahoma, School of Computer Science, Norman, OK 73019, ggruenwald@ou.edu
[2]Universidad Surcolombiana, matildemm@usco.edu.co

Key words
Mobile computing, real-time database, transaction management, ad-hoc netwroks

Abstract

A Mobile Ad-hoc Network (MANET) is an autonomous system of mobile hosts (MHs) with similar transmission power and computation capabilities that communicate over relatively bandwidth constrained wireless links. Applications such as emergency/rescue operations, conferences/meetings/lectures, disaster relief efforts, bluetooth (Personal Area Network) and military networks can be conceived as applications of MANET due to the fact that they cannot rely on centralized and organized connectivity. In these environment transactions are time-critical and require to be executed not only correctly but also within their deadlines, that is, the user that submit a transaction would like it to be completed before a certain time in the future.
This study focuses on the comparison of four scheduling techniques based on the policy of assigning priorities to transactions on the system. The techniques are: First Come First Serve (FCFS) [1,2], Earliest Deadline (ED) [1,2,5], Least Slack (LS) [1,2,8] and Least Slack Mobile (LSM) proposed in [3] where some modifications to the Least Slack Technique with respect to energy constraints, disconnection and transaction type (firm/soft) are considered. Applying these modifications to Earliest Deadline, the performance of the system will be evaluated to measure the percentage of transaction missing deadlines and the total energy consumption in the mobile hosts.
The performance evaluation of the techniques will be carried out by means of simulation. The simulation model is implemented using Visual Slam/Awesim [7].

Key words

Computación móvil, bases de datos en tiempo real, manejo de transecciones y redes ad hoc

RESUMEN

Una red Mobile Ad-hoc (MANET) es un sistema autónomo de servidores mobiles (MHs) con capacidades computacionales similares que se comunican entre si a través de una red wireless. Se consideran como aplicaciones MANET las operaciones de emergencia/rescate, conferencias/reuniones/clases magistrales, ayuda en caso de desastres, bluetooth (Personal Area Network) y redes militares, debido a que estos sistemas no pueden contar con una conexión centralizada y directa de los diferentes servidores que interactúan en el proceso. En este tipo de redes las transacciones son dependientes directamente del tiempo y requieren ser ejecutadas antes que expire el tiempo asignado para su ejecución, por lo tanto, el usuario que requiere la ejecución de una transacción espera que esta sea completada antes de un cierto tiempo en el futuro.

Este estudio se centra en la comparación de cuatro técnicas de programación de tareas (scheduling techniques) basadas en la política de asignación de las mismas de acuerdo con las prioridades que tengan dentro del sistema. Estas técnicas son: First Come First Serve (FCFS) [1,2], Earliest Deadline (ED) [1,2,5], Least Slack (LS) [1,2,8] and Least Slack Mobile (LSM) propuestas en [3], en donde se citan estas técnicas como trabajo futuro de la Investigación realizada por Mr. Banik Shankar, "Energy-Efficient Transaction Management for Real-Time Mobile Databases in Ad-hoc Network Environments" el cual fue presentado como trabajo de grado para optar el titulo de Master of Science en la Universidad de Oklahoma. Por tanto, esta investigación es continuación del trabajo hecho por Mr. Shankar. Aqui se presentan algunas modificaciones a la técnica de Least Slack con respecto a las limitaciones de energia, desconexiones y tipo de transacciones (firm/soft). Tales modificaciones son aplicadas tambien a la técnica de Earliest Deadline para evaluar el comportamiento del sistema y medir el porcentaje de transacciones que no alcanzan a ser ejecutados dentro del tiempo límite y para medir la cantidad de energía consumida por los servidores mobiles.
La evaluación del sistema se realizó usando simulación. El modelo de simulación es implementado usando Visual Slam/Awesim [7].

INTRODUCTION

In a Mobile Ad-hoc Network (MANET) there is no static infrastructure such as base stations. In an ad hoc network each node acts as a potential router, routing packets for two communicating nodes that may not be radio contact with each other. If two hosts are not within radio range, all message communication between them must pass through one or more intermediate hosts. Thus communication may be via multiple wireless hops. The hosts are free to move around randomly, thus changing the network topology dynamically. Applications such as emergency/rescue operations, conferences/meetings/lectures, disaster relief efforts, bluetooth (Personal Area Network) and military networks can be conceived as applications of MANET due to the fact that they cannot rely on centralized and organized connectivity.

Mobile MultiDatabase Management Systems (MMDBMSs) provide services to mobile users to access databases conveniently and efficiently [3]. A scheduling algorithm provides a set of rules that determine the processor to be used and the transactions to be executed at any particular point in time. In our real-time environment, transactions have deadlines and are classified into two categories: firm and soft [9]. Firm transactions must be aborted if they miss their deadlines while soft transactions still can be executed after their deadlines have expired.

This work intends to use priority scheduling techniques such that a higher-priority request has precedence over a lower-priority request to manage transactions in a Real-Time Mobile Database System in MANET based on deadline, energy constraints, disconnection and transaction type (firm/soft). The goal is to reduce the battery consumption and the percentage of transaction missing its deadline.

MOBILE AD HOC NETWORK ARCHITECTURE

The proposed Architecture in [3] for a Mobile Ad-hoc Network

(MANET) is illustrated en Figure 1 [3].

"Mhs in this architecture can be classified into two groups: 1) computers with reduced memory, storage, power and computing capabilities, which we will call Small Mobile Hosts (SMHs), and 2) classical workstations equipped with more storage, power, communication and computing facilities than SMHs, which we will call Large Mobile Host (LMHs). Every MH has a radius of influence. A MH can directly communicate with other MHs which are within its radius of influence. In Figure 1, an oval shape with borders in dotted line represents the radius of influence of a MH. The communication link between two MHs is shown with dark dotted lines. In this architecture, two MHs that are outside each other's radius of influence will be able to indirectly communicate with each other in multiple hops using other intermediate MHs between them [10]. For example, in Figure 1, SMH 11 will not be able to communicate directly with LMH 3 because their radii of influence are not overlapping, but it can indirectly communicate in multiple hops using SMH 10 and SMH 9 between them. Due to energy and storage limitations, we assume that only LMHs will store the whole Data Base Management System (DBMS) and SMHs will store only some modules of the DBMS (e.g. Query Processor) that allow them to query their own data, submit transactions to LMHs and receive the results. We also assume that database is distributed but not replicated."

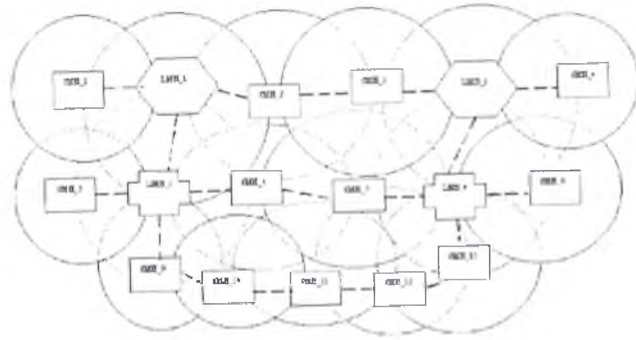

Figure 1. Proposed Ad hoc Network Architecture

As described in [3] supporting database transaction services in an ad-hoc mobile network raises new issues. These are as follow:

- A MH that holds a database, will submit/request data for processing of transactions to/from other MHs.
- Before a transaction is submitted to another MH a route must be found since in this environment both the user and the data source will be moving. Thus, the Transaction Manager at the MH where the database is stored has to consider the mobility of the submitting MHs as well as the deadlines of the transactions.
- Another important issue in ad-hoc networks is power or energy restriction on MHs because MHs are not connected to direct power supplies and many of them will run on small and low-power devices. Therefore, energy-efficient solutions are needed for this environment. Such solutions should aim to provide a balance of energy consumption among MHs so that MHs with low energy do not run out of energy quickly, and thus the number of MH disconnections can be reduced.

TRANSACTION MANAGAMENT IN MANET

As proposed in [3], Each MH stores the following information in its local database.

- The ID field which uniquely identifies a MH.
- The Position of the MH which obtains its coordinates from GPS (Global Positioning Scheme) [11] periodically. This position information is used at the time of routing a transaction from a source MH to a destination MH.
- The Radius of transmission range.
- The Energy_availability which records the amount of energy available at that time. This information is needed to identify the LMH with the highest available energy to submit a transaction and to identify the SMH with less energy to give the higher priority in the case of competing for processing with equal deadline or slack time.

In addition each LMH will maintain a Global Schema, which is the integration of all local schemas from all LMHs. It will also store the corresponding ID of the LMH for each local schema. This Global Schema is required to identify which data object is stored in which LMH.

In our environment, a global transaction is defined as a transaction, which requires data items from different sites. The part of a global transaction, which is executed in a particular site, is defined as a sub-transaction. It is assumed that for a particular global transaction there will be one sub-transaction for one site.

The SMH will initiate a global transaction and submit it to an LMH. This LMH will act as a coordinator for this global transaction. Then the LMH coordinator will check the global schema to find which data item is stored at which LMH and will divide the global transaction into sub-transactions. These LMHs are the participant sites for the global transaction. The LMH coordinator will submit the sub-transactions of the global transaction to the respective LMH participants. Then, the LMH coordinator with the help of the LMH participants will commit/abort the global transaction and will return the result to the requesting SMH.

In order to provide a balance of energy consumption among MHs and to reduce the number of transactions that must be aborted due to miss their deadlines the SMH must submit their firm transactions to the nearest LMH and their soft transactions to the LMH with the highest energy available. Here, we are sacrificing the first deadlines of soft transactions in favor of balancing the energy consumption because soft transactions can still be executed after their first deadlines have expire [3].

A LMH can receive two types of transactions: global transactions from an SMH or sub-transactions from other LMHs (called Participant sites). Each LMH has three parts:

- Transaction Scheduler (TS): schedules all global transactions and sub-transactions.
- Transaction Coordinator (TC): divides the global transaction into sub-transactions and submits them to corresponding LMHs, and returns the results to the requesting MH.
- Transaction Manager (TM): manages the execution of sub-transactions.

The TS at LMH will use one of the real-time energy-efficient dynamic scheduling algorithms described in section 4 to schedule transactions. Transactions will be organized in a queue that reflects its priorities.

SCHEDULING TECHNIQUES

When a LMH receives a transaction from a SMHs or other LMHs, it has to assign priorities among transactions in order to schedule them. In our environment the scheduling algorithm has to consider not only transaction types (firm and soft), transaction deadlines, but also the energy limitations of the MHs. In order to handle the above considerations the following algorithm will be used:

Begin

Calculate the deadline/slack time for all transactions using Equation 1, 2 or 3.

Sort all the transactions according to their deadlines/slack times.

Assign higher priorities to transactions with shorter deadlines/slack times.

If two firm transactions or two soft transactions have the same deadline/slack time

Then give priority to the one whose requesting MHs have less energy.

If two transactions have equal energy

Then give priority to the one that arrives first

End if

End if

If the deadline/slack time of a firm transaction is equal to the deadline/slack time of a soft transaction

Then give a higher priority to the firm transaction.

End if

End

In our environment when a transaction arrives to the system,

the following information will be known:

Release time: It is the earliest time the transaction can be

started and is usually the arrival time (AT) of the transaction.

- Deadline (D): It is the desired maximum commit time.
- Runtime Estimate (RE): It approximates the duration of the transaction on an unloaded system. It takes into account both the CPU and disk access times.
- Slack Factor (SF): It is a constant value that determines the tightness/slackness of deadlines.
- Slack Time (S): It is the maximum amount of time that a transaction can spend without executing and still complete within its deadline.

First Come First Serve Technique (FCFS)

As described in [1], This policy assigns the highest priority to the transaction with the earliest release time. If release times equal arrival times then we have the traditional version of FCFS. The primary weakness of FCFS is that it does not make use of deadline information. FCFS will discriminate against a newly arrived task with an urgent deadline in favor of an older task that may not have such an urgent deadline.

Earliest Deadline (ED)

As described in [1], The transaction with the earliest deadline has the highest priority. A major weakness of this policy is that it can assign the highest priority to a task that already has missed or is about to miss its deadline. By assigning a high priority and system resources to a transaction that will miss its deadline anyway, we deny resources to transactions that still have a chance to meet their deadlines and cause them to be late as well.

Least Slack (LS)

This technique is based in equation (2) [1,2,8]. In the LS technique, transactions with less slack time are scheduled before transactions with more slack time. Figure 2 explains the principle of equations 1 and 2.

$$D = AT + RE * SF \quad (1)$$

$$S = D - (AT + RE) \quad (2)$$

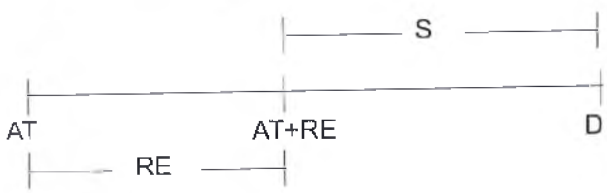

Figure 2. Sketch of the principle of equations 1 and 2

Least Slack Mobile (proposed in [3]) (LSS)

The slack time, S, proposed in [3] to schedule a transaction could be calculated using the following equation:

$$S = D - (AT + RE + P_d * T_d) \quad (3)$$

where 'D' is the deadline, 'AT' is the current time (current time is equal to arrival time since the slack time is calculated when transaction arrives to the system), 'RE' is the runtime estimate, 'Pd' is the probability of disconnection during execution and 'Td' is the average time loss due to disconnection.

SIMULATION MODEL AND RESULTS

Description of the simulation model

The simulation model is implemented using Visual Slam/Awesim [7]. Global transactions are defined as entities in the simulation model. The attributes associated with each transaction are Transaction Creation Time, Transaction ID, Number of site transactions, Number of operations per site transactions, Runtime Estimate, Deadline, slack time, Transaction Type (firm or soft) and ID of the SMH which initiates the transaction.

The mobile hosts are defined as resources in the simulation model. The attributes associated with each resource are Resource ID, Position which consists of X coordinate and Y coordinate, Energy Level, Radius of Influence, Mode (Active, Doze or Sleep), Time in Active Mode, Time in Doze Mode and Time in Sleep Mode.

To evaluate the performance of the system using the scheduling techniques mentioned above two-performance metrics are considered: the percentage of transactions missing their deadlines and energy consumption of mobile hosts. The equation for % Missed deadline is as follows:

$$\%Missed \quad deadline = 100 * \frac{\#of \ transactions \ missed \ dealine}{\#of \ transactions \ processed}$$

The energy consumption for a resource is obtained by multiplying the power of the resource with the length of the time the resource was in the active/doze mode to process transactions.

Energy consumption of a resource = (Power of the resource in the active mode) * (The time the resource was in the active mode) + (Power of the resource in the doze mode)* (The time the resource was in the doze mode)
(5)

Simulation Parameters

The environment of the simulation is assumed to be a closed area of 10001000 units in which the initial positions (coordinate X, coordinate Y) of all the mobile hosts have been randomly distributed. The parameters are classified as static and dynamic parameters. These parameters are summarized in Table 1 and 2 [3].

Table 1. Static Parameters of the Simulation Model

Parameter	Meaning	Default Value
Bandwidth	Bandwidth of wireless medium	100 kbps
CPU_power_LMH	CPU Power of LMH	140 MIPS
CPU_power_SMH	CPU Power of SMH	4 MIPS
Et_tran	End transaction	0.0054 ms
LMH_power_rate	LMH Power Dissipation Rate	170W per hour
Mem_access_time	Main Memory access time per word	0.00018 ms
Num_ops	Number of operations per site transaction	UNIF (5,10)
Num_site_tran	Number of site transactions in a global transaction	TRIAG(3,4,5)
Pre_op	Preprocess one operation	0.000007 ms
Pre_trans	Preprocess one transaction	0.0072 ms
Prob_read	Probability of read	60%
SMH_power_rate	SMH Power Dissipation Rate	7W per hour
Word_size	Number of bytes per word	8
Radius_SMH	Radius of influence of SMH	100 units
Radius_LMH	Radius of influence of LMH	200 units

Table 2. Dynamic Parameters of the Simulation Model
*Indicates the parameter that have been modified from [3]

Parameter	Meaning	Default Value	Range
IAT	Inter Arrival Time between global transactions	EXPON (100)*	EXPON(25) to EXPON(100)*
Firm_Prob	Probability that a transaction is Firm	0.5	0.1-1
Num_SMH	Number of SMH	40	20-40
Num_LMH	Number of LMH	20	5-20
Slack_factor	Slack factor	10*	10-25*

Experiments

In this section, we describe different sets of experiments and analyze their results to evaluate the performance of our techniques. For each set of experiments, we run the simulation with Inter Arrival Time of 100 ms and slack factor of 10. These values were chosen because the experiments of varying the Inter Arrival Time on percentage Missing Deadlines for All Firm and Soft Transactions (see Figures 16,17) show the best results. Each run continues until a total of 1000 transactions are completed in the system.

The impact of disconnection was evaluated by using the technique proposed in [3]. Figure 3, shows that the percentage of transaction missing deadline increases with the probability of disconnection for All Firm and Soft Transactions. In order to implement equation 3 to run for LSM technique, an average value for probability of disconnection of 0.4 was assumed for a period of 5 seconds [13].

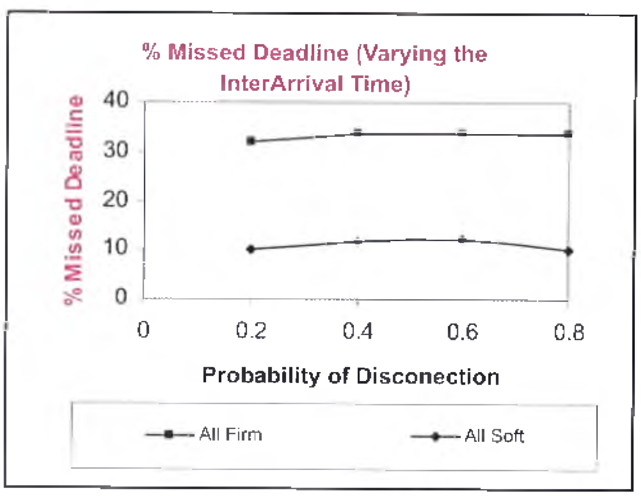

Figure 3. Varying the Probability of Disconnection on % Missed Deadline for All Firm and Soft Transactions

Algorithm FCFS misses a greater number of deadlines for all the runs. This technique is included in this study for comparison with scheduling techniques that gives higher priority to transactions with least deadline/slack time. It can be observed from all the experiments that these techniques perform better because they allow transactions with more urgent deadline/slack time to preempt transactions with less urgent deadline/slack times.

Varying the Number of LMHs

In this experiment, we have varied the number of LMHs for Firm and Soft transactions. From Figure 4 and 5, we can observe that the percentage of transactions missing deadline decreases as the number of LMHs increases for all scheduling techniques. The LMHs in our environment are servers; then with more servers available the transactions will spend less time to be processed, hence fewer transactions missing its deadlines. For all Soft transactions, the percentage of transactions missing deadline is lower than for Firm transactions. This is due to the fact that soft transactions are aborted when they miss its second deadline, which is twice of their first deadline. Earliest Deadline performs better than the two Least Slack techniques because as shown in Figure 16 and 17 this technique performs better at lower load of the system. Here we are running for Inter Arrival time of 100 ms which is the lower load of the system.

The results can also be explained by analyzing the equations 2 and 3 to determine the priority, or slack time. We can notice that with LS priority assignment there is no correlation between the Arrival Time of the transactions and their slack time. By the contrary, with Earliest Deadline there is a direct correlation. Thus, a transaction T_2 that arrives much later than a transaction T_1 is more likely to have a deadline that is greater than the one for T_1 Then it is less likely that transaction T_2 will preempt transaction T_1. These can be clear by the following example. Transaction T_1 arrives at time 10, has a slack time of 5 and a deadline of 20. Transaction T_2 arrives a time 20, has a slack time of 4 and a deadline of 28. If we used LS to schedule the transactions then T_2 will preempt transaction T_1 since it has least slack time. However, if we used ED, transaction T_2 will not preempt T_1 since its deadline is higher.

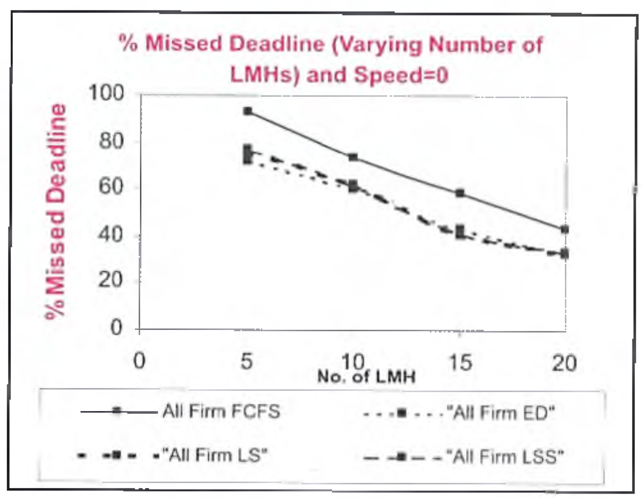

Figure 4. Varying LMHs on % Missed Deadline for All Firm Transactions

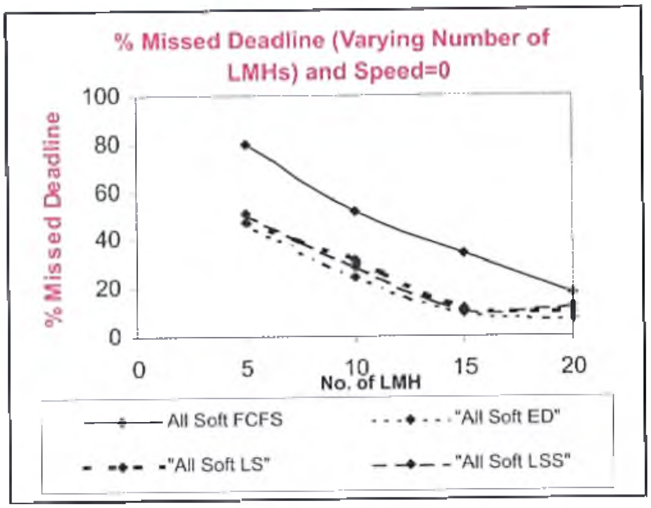

Figure 5. Varying LMHs on % Missed Deadline for All Soft Transactions

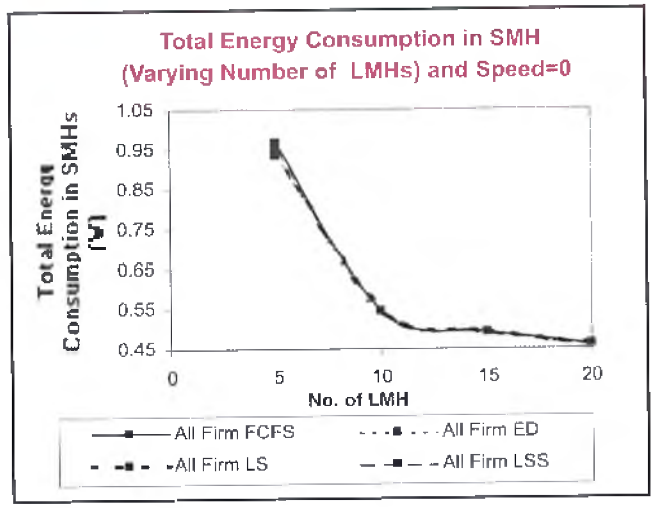

Figure 8. Varying LMHs on Total Energy Consumption in SMHs for All Firm Transactions

It can be observed from Figure 6, 7, 8 and 9 that the total energy consumption for All Soft and All Firm Transactions in LMHs and SMHs increases as the number of LMH increases. This is because when having fewer transactions that missed their deadlines more energy needs to be consumed in to order to process them. From Figure 7 it can be observed that for all Soft transactions the energy consumed is higher compared to All Firm transactions since the percentage of transactions missing deadline is less for soft transaction. It also can be seen that, since ED has fewer transactions missing deadline, it consumed more energy.

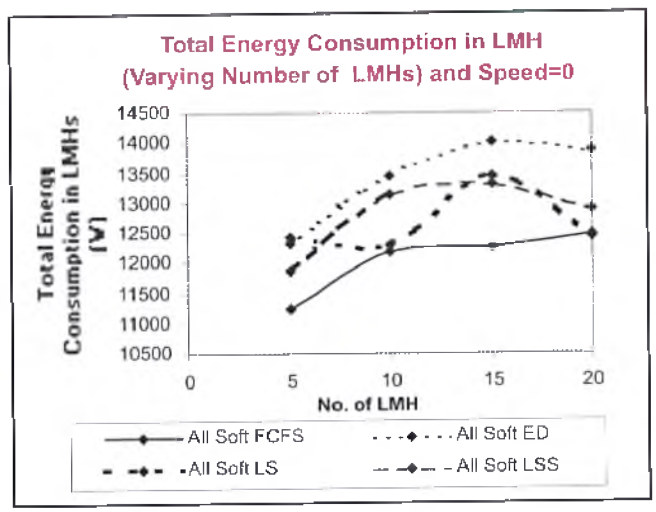

Figure 7. Varying LMHs on Total Energy Consumption in LMHs for All Soft Transactions

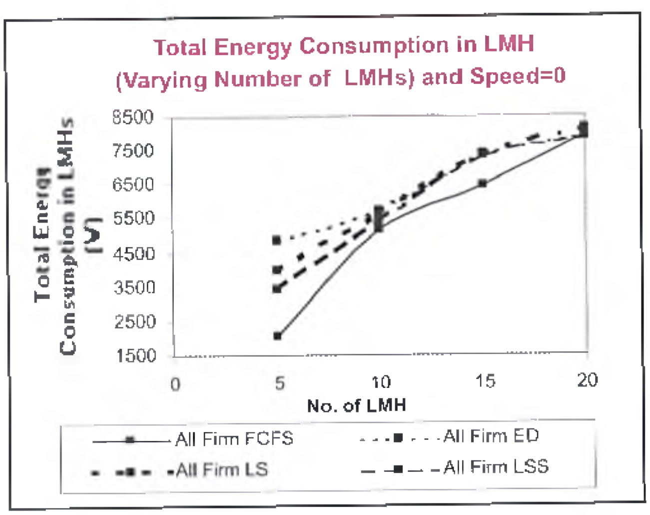

Figure 6. Varying LMHs on Total Energy Consumption in LMHs for All Firm Transactions

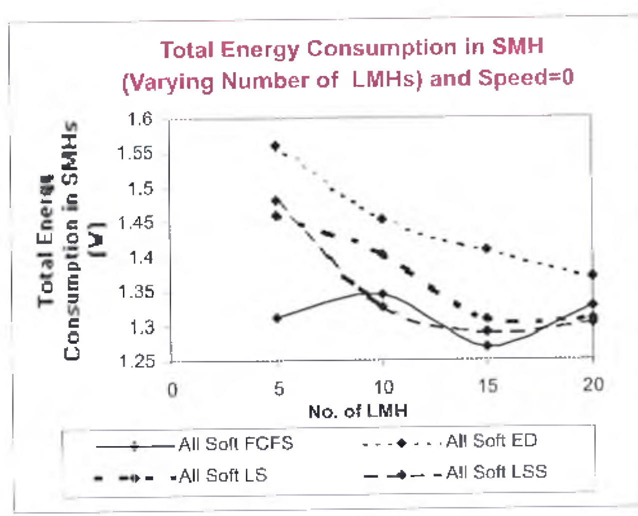

Figure 9. Varying LMHs on Total Energy Consumption in SMHs for All Soft Transactions

Changing the Speed of MHs

Mobile hosts in our system can move randomly in one of the eight directions (↗,↙,↖,↘,←,→,↑,↓). Experiments changing the speed of mobile hosts were carried out for different speeds on mobile hosts starting with the initial position and moving with the same random moving pattern. From Figures 10 and 11, changing the speed of mobile hosts has negligible effects on the percentage of transaction missing deadline for all the techniques. The overall percentage of transaction missing deadline is explained with the overall average distance. When the transaction has a higher percentage of missing deadline, shorter distance is covered by the transaction. For example, All Firm Transaction has a higher percentage of transaction missing deadline and a shorter overall average distance. The average distance covered by the transaction shows no significant change with speeds. In the simulation model, the overall average distance is affected by two factors. First, some mobile hosts move closer to each other whereas other mobile hosts move further in the system. As a result, the overall average distance remains almost the same for different speeds. Second, a transaction with a shorter deadline covers a shorter distance or vice versa [12]. ED technique performs better than LS techniques.

From Figures 12, 13, 14 and 15 the energy consumed by LMHs and SMHs for All Soft and All Firm transactions show that ED spend more energy. This is because ED is the technique that misses fewer deadlines. Since the percentage of transaction missing deadline shows no significant variation as the speed changes, it is also applied to the energy consumed by the mobile hosts. If more transactions miss the deadline, it implies fewer transactions to be processed by the LMHs, thus less energy consumed [12]. For LMHs and SMHs, the energy consumption is high when all the transactions are soft.

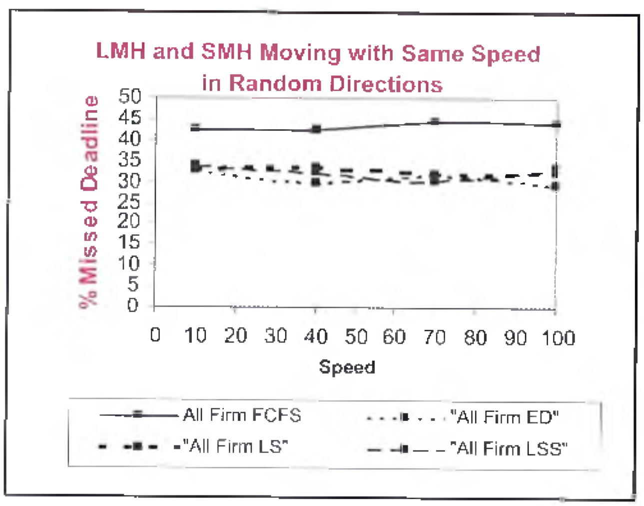

Figure 10. Changing Speed on % Missed Deadline for All Firm Transactions

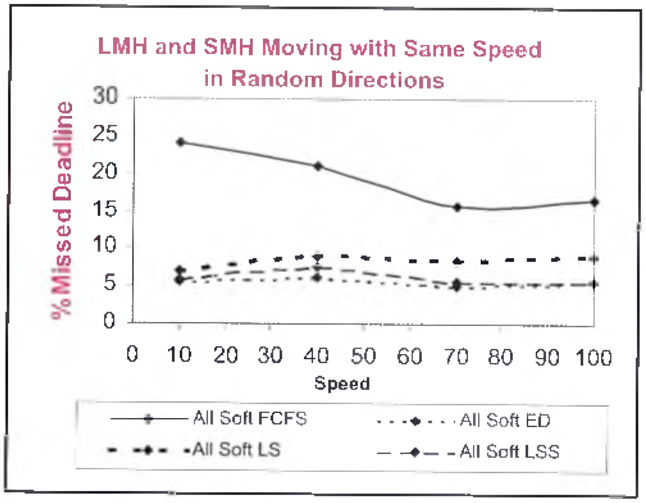

Figure 11. Changing Speed on % Missed Deadline for All Soft Transactions

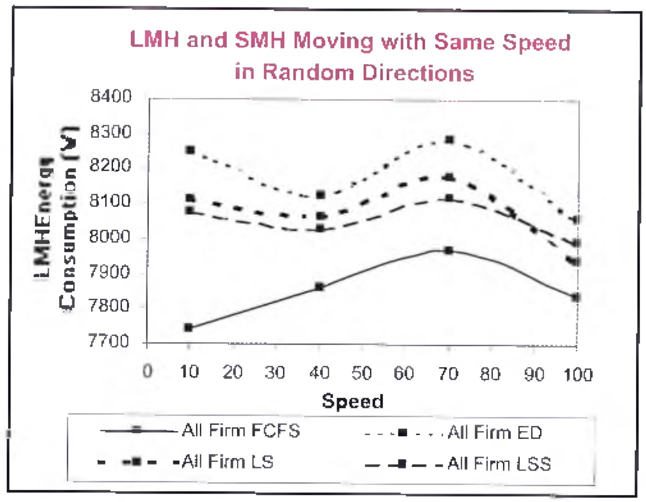

Figure 12. Changing Speed on Total Energy Consumption in LMHs for All Firm Transactions

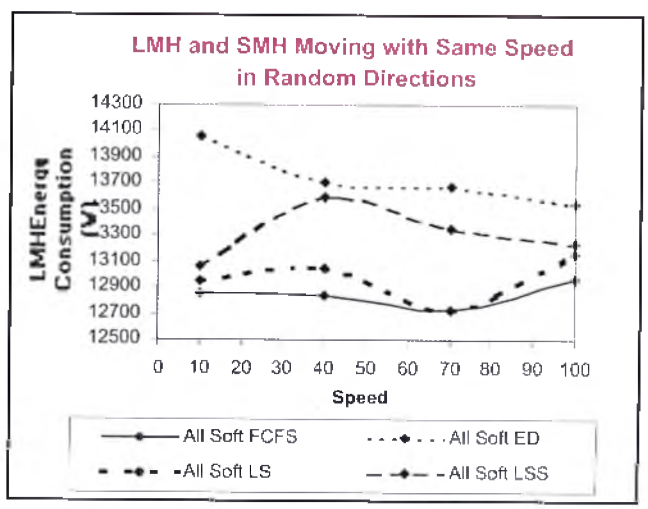

Figure 13. Changing Speed on Total Energy Consumption in LMHs for All Soft Transactions

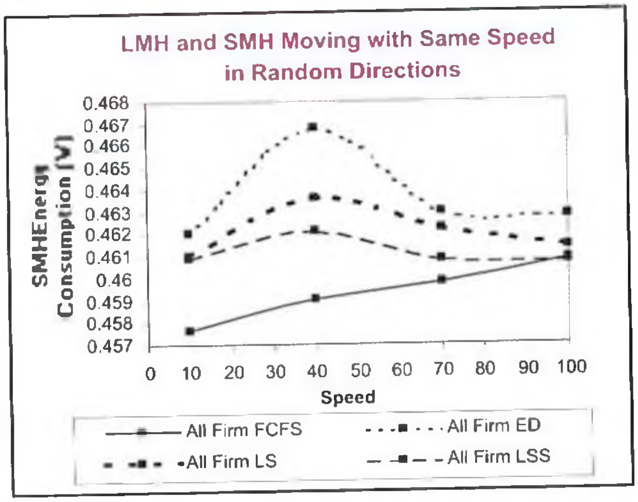

Figure 14. Changing Speed on Total Energy Consumption in SMHs for All Firm Transactions

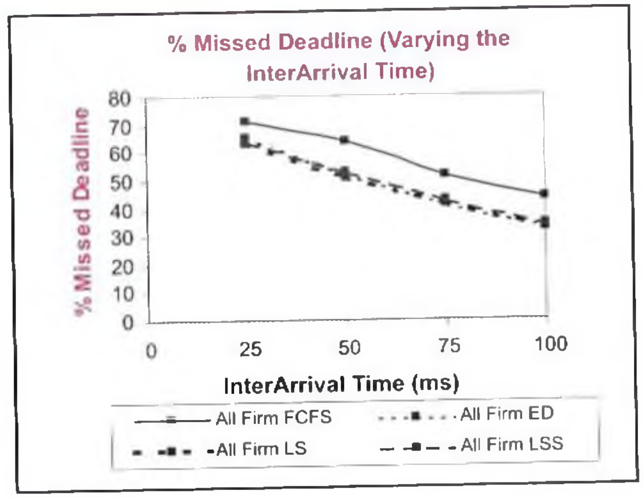

Figure 16. Varying the Inter Arrival Time on % Missed Deadline for All Firm Transactions

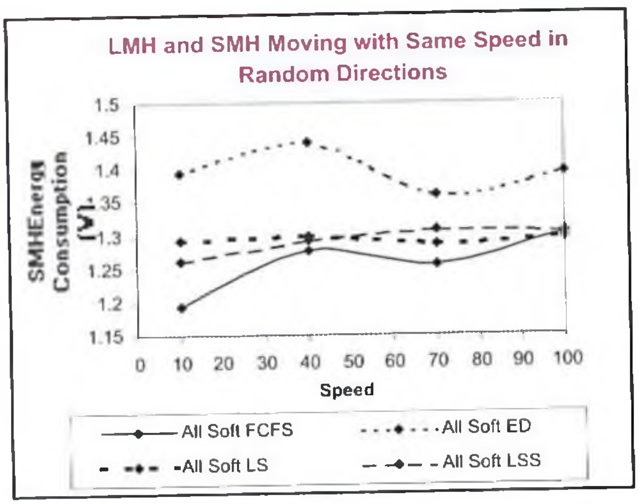

Figure 15. Changing Speed on Total Energy Consumption in SMHs for All Soft Transactions

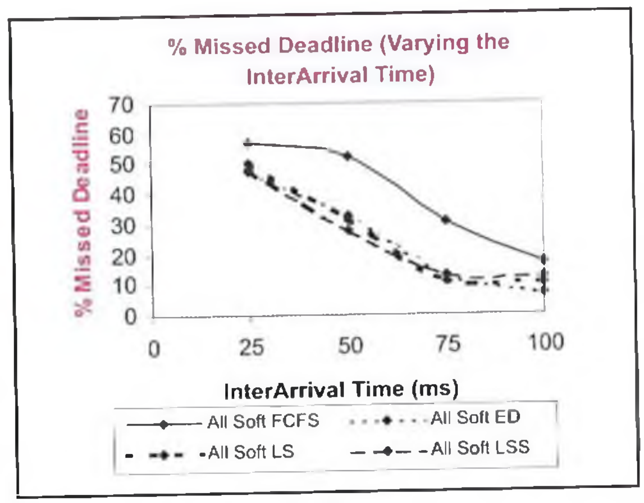

Figure 17. Varying the Inter Arrival Time on % Missed De adline for All Soft Transactions

Varying the Inter Arrival Time

The Inter Arrival Time of global transactions is exponentially distributed with mean between 25 and 100 time units. In this experiment, we varied the Inter Arrival time of Firm and Soft transactions to study the effect of workload on the system. It can be observed from Figures 16 and 17 that the percentage of transactions missing deadlines decreases as the Inter Arrival Time increases. This is because when the Inter Arrival Time of transactions is increased, fewer transactions enter into the system to be processed decreasing the system load.

Figures 16 and 17, also show that the Earliest Deadline technique performs better than Least Slack at lower load of the system. ED performs poorly at higher load because it assigns high priorities to transactions that have missed or about to missed their deadlines.

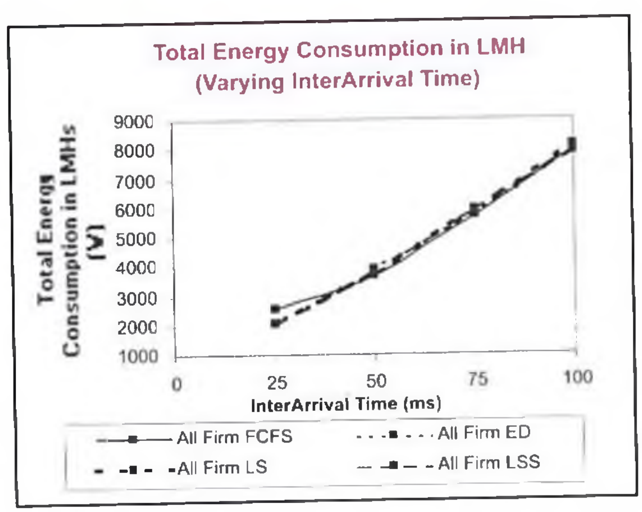

Figure 18. Varying the Inter Arrival Time on Total Energy Consumption in LMHs for All Firm Transactions

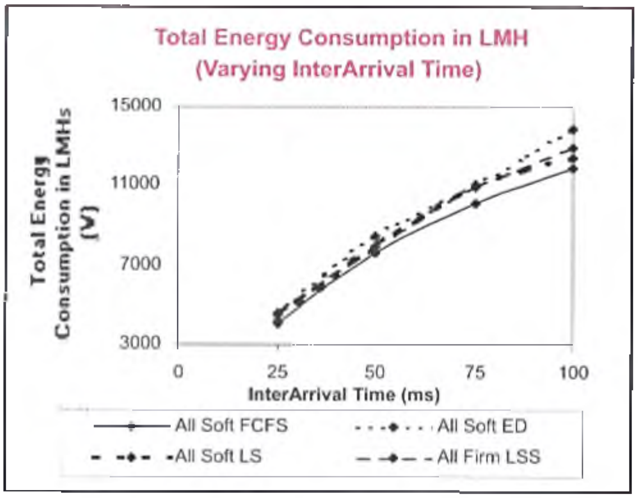

Figure 19. Varying the Inter Arrival Time on Total Energy Consumption in LMHs for All Soft Transactions

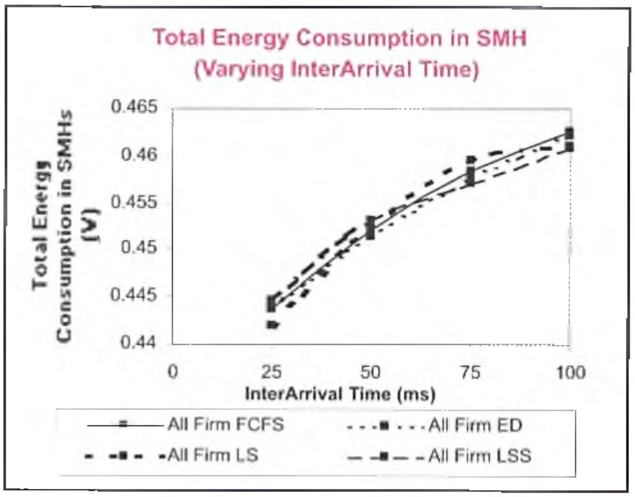

Figure 20. Varying the Inter Arrival Time on Total Energy Consumption in SMHs for All Firm Transactions

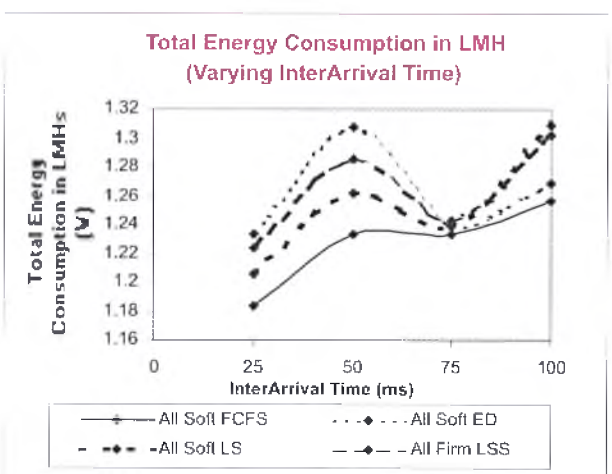

Figure 21. Varying the Inter Arrival Time on Total Energy Consumption in SMHs for All Soft Transactions

From Figures 18,19, 20 and 21 we can see that energy consumption of LMHs and SMHs increases with the Inter Arrival Time. This is because if more transactions finish before their deadline they need more energy to be process. As mentioned earlier, transactions executing under ED policy have least percentage of transactions missing deadline, thus more energy will be consumed.

For every mobile host in the system, we varied the Inter Arrival Time for Soft Transactions and calculate the energy consumed to check how the energy is distributed among them. The results for all the scheduling techniques, for earliest deadline only and for Earliest Deadline and Least Slack Combined show that if the Inter Arrival Time of transactions is large, i.e. the system load is low, the energy consumed is more evenly distributed among the mobile hosts.

CONCLUSIONS AND FUTURE RESEARCH

The following was observed from this study:

- For all the tested techniques to manage transactions in Real-Time Database systems in MANET when missing deadline, we observed that ED performs better. LS show better results when working at high load of the system.
- When increasing the number of LMHs for all techniques, the load of the system decreases and hence the percentage of transactions missing deadlines decreases.
- The energy consumption for all the techniques increases when increasing the LMHs in the system due to the fact that fewer transactions miss their deadlines and then need to consume more energy.
- Since All Firm transactions miss more deadlines than All Soft transactions the energy consumed is less compared to the All Soft.

The technique that we used to run our experiments as described in [3] is called Transaction Type Based Server Assignment (TTBSA). Here the LHM that is assigned to handle transactions initiated by SMHs, always submits firm transactions to the nearest LMH and soft transactions to the LMH with the highest energy. The goal is to reduce the number of transactions missing their deadlines as well as to balance the energy consumption among the LMHs in the system.

As an extension of this research work, we recommend to perform a set of experiments to compare two more alternatives proposed in [3] to handle the assignment of LMH to process the transactions for all the scheduling techniques analyzed here. These techniques are called Location Based Server Assignment (LBSA) and Energy Based Server Assignment (EBSA). The first considers only LMH location and the other considers only LMH energy. In LBSA, all transactions initiated by SMHs will always be sent to their nearest LMHs. In EBSA, all transactions initiated by SMHs will always be sent to the LMH with the highest energy.

Future research can also test the techniques examined here after implementing the Sleep mode of mobile hosts in the system.

REFERENCES

1. Abbott R., H. Garcia-Molina, "Scheduling Real-Time Transactions: A Performance Evaluation". ACM Transactions on Database Systems, Volumen 17, No 3, September 1992, Pages 513-560.

2. Abbott R., H. Garcia-Molina, "Scheduling Real-Time Transactions: A Performance Evaluation". In Proceedings of the 14th VLDB Conference. Los Angeles, Aug 29- Sep 1, 1988. Pages 1-12.

3. Banik Shankar."Energy-Efficient Transaction Management for Real-Time Mobile Databases in Ad-hoc Network Environments". Master Thesis, The University of Oklahoma, 2000.

4. Gruenwald L., Banik S. "Energy Efficient Transaction Management for Real-Time Mobile Databases in Ad hoc Network Environments". 2nd International Conference on Mobile Data Management, Hong Kong, January 2001. Pages 287-288.

5. Huang, J.; Stankovic, J.A.; Towsley, D.; Ramamritham, K. "Experimental evaluation of real-time transaction processing". Real Time Systems Symposium, 1989., Proceedings.,1989,Page(s): 144–153.

6. Imielinski Tomasz., Banadrinath B.R. "Mobile Wireless Computing : Solutions and Challenges in Data Management". Communications of the ACM(CACM), Vol. 37, October 1994, Pages 18-28.

7. Prisker A. Alan B, O'Reilly Jean J. "Simulation with Visual SLAM and Awesim". System Publishing Corporation, 1999.

8. Brad Adelberg, Hector Garcia Molina, Ben Kao. Emulating Soft-Real Time Scheduling Using Traditional System Schedulers. April 25, 1994.

9. Gruenwald, L., et al., "Database Research at The University of Oklahoma", ACM SIGMOD RECORD, Vol. 28, No. 3, September 1999, Pages 73-78.

10. Bandyopadhyay, S., and K. Paul, "Evaluating the Performance of Mobile Agent-Based Communication among Mobile Hosts in Large Ad-Hoc Wireless Network", Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 1999, Pages 69-73.

11. Ko, Y., N. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad-Hoc Networks", MOBICOM 1998, Pages 66-75.

12. Gruenwald L, Shankar M. Banik, Chuo Ning Lau, Matilde Montealegre. Managing Real-Time Database Transactions in Mobile Ad-Hoc Networks. This is a paper submitted for publication. Chuo Ning Lau and Matilde Montealegre contributed with the implementation of the speed at which the mobile hosts move. A special study at the University of Oklahoma, summer 2001.

13. Ersan Kayan, Ozgur Ulusoy. Real-Time Transaction Management in Mobile Computing Systems. Department of Computer Engineering and Information Science. Bilkent University. Bilkent, Ankara 06533, Turkey.