Load Balancing: An Approach Based on Clustering in Ad Hoc Networks

Rachida Aoudjit[1], Mustapha Lalam[1], Abdelaziz M'zoughi[2], Malika Belkadi[1] and Mehammed Daoui[1]

[1]Université de Tizi-Ouzou, Département d'Informatique, Algérie
[2]Institut de Recherche en Informatique de Toulouse, UPS, France

Mobile ad hoc networks consist of freely moving nodes responsible of not only forwarding packets for other nodes but can also perform extensive computations. One of the most critical issues in these networks is the significant differences in term of processing and energy capacity between the nodes, inducing a load imbalance. Thus, sharing the load between the overloaded and idle nodes is a necessity in ad hoc networks. In this paper, we present a new load balancing algorithm based on clustering where a subset of nodes 'clusterheads' is elected to maintain some balance within their respective clusters while minimizing the overall communication cost. Our primary goal is to minimize the total execution time of the tasks by distributing the workload among nodes. Another goal is to extend the overloaded nodes lifetime inducing a stability of the network. The simulation results have shown that network performance can be reached by distributing load to idle nodes within the network.

Keywords: ad hoc networks, load balancing, clustering, stability and energy conservation

1. Introduction

With the proliferation of mobile processing platforms and small sized wireless equipments, ad hoc networks have gained more attention these last decades. Ad hoc networks consist of a number of nodes having different computation and communication capabilities. They are equipped with small batteries. This puts significant constraints on the power available for computation and communications. Several researches have been undertaken to find efficient means to preserve this precious resource [21]. Otherwise, in an ad hoc network, with a decentralized and heterogeneous structure, some nodes may have different capabilities of processing and batteries. In such environments, imbalances of load can occur. Indeed, a more powerful node in term of processing capacity can become idle, because it has finished its work quickly while the others, less powerful, are occupied most of the time, consuming more energy. Powerful nodes capacity can be exploited by overloaded nodes if a fraction of their load is shared with them. If the difference between the heaviest loaded and the lightest loaded nodes is minimized, the average work execution time can be reduced, the energy of the nodes will be better exploited and the nodes lifetime can be extended. It is what contributes to the stability of the network topology that plays a principal role in different problems like: routing, scheduling, resource reservation [10]etc. Load balancing is certainly one of the solutions for increasing the efficiency of applications and the network life time. Load Balancing algorithms are designed essentially to distribute equally the load on nodes and maximize their utilization while minimizing the total task execution time. This issue has been of considerable interest in the network research community when it comes to wired [29, 30, 2, 26, 3, 17] and wireless [24, 13, 22, 23, 20] networks. It aims to guarantee that no node is underloaded or overloaded. It looks at setting up a uniform load on all nodes. Then, it is expanded in order to take into account new environments and new applications (large scale applications, multimedia applications, etc.). Compared to the wired networks, the mobile environments introduce new highly variable parameters such as limited resources, wireless link communication and mobility. In this work, we propose a new approach for load balancing in MANETs which remains an NP complete problem. It is based on the clustering which organizes the nodes into clusters, where some nodes work as clusterheads and coordinate other nodes in the clusters. Clustering has advantages such as improving bandwidth utilization by reducing communications overhead and reducing energy consumption. Our algorithm purpose is to reduce the execution time and maximize the lifetime of the network as much as possible while minimizing the overall communications cost. The remainder of this paper is organized as follows: Section 2 presents the significant works done and related to the load balancing issue and our motivation, followed by the description of the proposed approach and its details in Section 3. The simulation environment and the simulation results interpretation are presented in Sections 4 and 5.

2. Previous Work and Motivation

Improving network performance by balancing the network load has been of considerable interest in the network research community. Existing load balancing mechanisms for ad hoc networks can be classified into two categories, according to the level to which they are applied: on the communication level or on the computing level.

Communication level: research in this area has been focused primarily on the construction of alternate route sets and implementation of policies for traffic distribution among these multiple routes. A special class of routing protocols for ad hoc networks deals with load balancing [22, 23, 13, 19, 27, 28].

Computing level: developing strategies in this area deal with distributing and mapping tasks to a computational resource in a way that balances the load, to reduce the total processing time and improve the node utilization. Few works dealing with the problem from this point of view can be cited in the literature.

Load balancing in [1, 14] is treated as an extension to clusterhead election. It allows all nodes an equal opportunity to be a clusterhead and extend the its duration based on some input parameters.

In [24], load balancing tries to find the best node to share the load. The paper provides interesting results, but a number of aspects are not covered in depth. For example, the overhead due to the communications and the energy consumption.

Otherwise, due to the nature completely distributed of the ad hoc networks and the absence of a centralized infrastructure, the control of its topology is difficult. If we divide the network into small zones with a centralized structure to simplify and control the topology, the load balancing problem becomes less complicated. In this case, each node has the information about the load status of all its neighbors through a coordinator node. One way to solve this problem is to group the nodes into clusters, where one node in each cluster functions as a clusterhead.

Knowing that a load balancing strategy achieves high performance if it produces a minimal overhead of communications because of the limited energy and bandwidth in mobile ad hoc networks, our solution is based on the concept of clustering where a subset of nodes 'clusterheads' is elected to coordinate their respective clusters [5, 12]. They can be used to control the load, to control the energy consumption and to serve as regional diffusers. The nodes registered with the nearest clusterhead become members of that cluster.

Our contribution in this work is to find the most suitable nodes to share the load for avoiding, or at least reducing, imbalances with a minimum of engendered overhead. Indeed, the algorithm that balances the loads should require little communications between the nodes, since mobile nodes are powered by batteries. The proposed algorithm takes into account the nodes localization; it is invoked each time that the imbalances occur by respecting a load threshold. Its performance is evaluated in terms of work execution time and nodes energy consumption.

3. The Proposed Approach

The idea of dividing a geographical area (ad hoc network) into small zones known as virtual cells is inspired by the working philosophy of cellular networks where the basic station is the principal coordinator at each cell level. This concept has been presented in the literature as clustering. Many clustering algorithms have been proposed. Most of them are heuristic in nature and their aim is to generate a minimum number of clusters [4, 8, 12, 6, 9, 18, 16]. They are distinguished by the different performance metrics taken into account and the criteria of the clusterhead selection. A thorough survey of ad hoc clustering protocols, as well as a performance based comparison among some of the most representative solutions, has been presented in [15, 7].

Our solution is presented in two steps:

1. Clusters formation: A subset of nodes 'clusterheads' is elected to coordinate their respective clusters.
2. Load balancing within each cluster: Our load balancing algorithm is a centralized algorithm since the global members load information of each cluster is collected by the clusterhead which ensures some balance between its members.

3.1. Clusters Formation

In our work, a clusterhead is elected for its relatively high energy capacity and its low mobility. Energy is a critical resource in ad hoc networks. A clusterhead consumes more energy than an ordinary node since it has other functionalities: coordination between its members, cluster maintenance and load balancing. Mobility is also an important factor for a clusterhead election. Indeed, to avoid frequent clusterheads changes, it is important to choose the one that has a low mobility. If the clusterhead moves quickly, the nodes can be detached from it, inducing re-affiliations which cause significant updated information exchanges.

3.1.1. Clusters Formation Procedure

1. Find the neighbors of each node i (nodes within its transmission range). They are defined as follows: $V(i) = \{i' \neq i / dist(i,i') \leq tx_{range}\}$ where tx_{range} is the transmission range of node i.
2. Compute the speed average for every node. This gives a measure of its mobility.
3. Compute the battery power for each node
4. Choose the node which has the smallest value of mobility and the highest value of battery power as clusterhead. All its neighbors are designated as its members and they can no longer participate in the election procedure.
5. Repeat steps 2-4 for the remaining nodes not yet assigned to any cluster.

The following characteristics are considered in the clusterhead election procedure:

- The nodes mobility causes changes in the network topology. If at a given time a node is detached from its current cluster and is attached to another, the corresponding clusterheads will update their members' tables.
- If a node leaves its cluster and doesn't find any other cluster to attach itself, the clusterheads election procedure is invoked.
- A clusterhead keeps the information concerning its members (identifier ID, status, load, energy). It can detect if another clusterhead is entered in its cluster. In these cases, one of them is constrained to give up its clusterhead's role. In our case, it is the one which has less energy.
- Because of the additional functionalities for which the clusterhead is intended, its energy is likely to be exhausted. A minimum threshold of energy is defined for each clusterhead. If it is reached, the clusterheads election procedure is invoked.

3.1.2. System Activation and Update Policy

When the network is set up, each node diffuses its identifier (ID) through a HELLO message which is recorded by all the other nodes in its transmission range. Once the neighbors list of each node is ready, the clustering procedure is executed.

Each node maintains a record of its status: (clusterhead or member). If it is not a clusterhead, it must know the clusterhead to which it is affiliated (CID: Clusterhead Id).

Considering the dynamic nature of the system, the nodes and the clusterheads tend to move in various directions, causing a disorganization of the network configuration. The system must be updated from time to time. The updates are made in two cases:

- When renewing cluster formation.
- When a node changes its affiliation from one clusterhead to a new one, in the existing clusterheads set; in this case, we speak about re-affiliation.

If a clusterhead doesn't receive any message from a node, it updates its members list. When a node is attached to another clusterhead, it updates its CID and the corresponding clusterhead updates its members list.When the clusterhead doesn't receive any more messages of a node, it updates its members list.

It is important to note that, in this work, we make some assumptions. The number of nodes per cluster is limited to avoid the clusterhead congestion and for better resources utilization [7, 11]. In all the analyses carried out on the proposed algorithm, we assumed that all nodes are cooperative and trusted. Considering these conditions allowed us to study only the characteristics specific to the proposed algorithm.

3.2. Load Balancing Algorithm at the Level of Each Cluster

As mentioned previously, a principal role of a clusterhead is the maintenance of load balancing in each cluster. It is the principal coordinator of its cluster; it collects periodically information about each member node of its cluster, such as energy and load values. These data are collected in the members table. When a node reaches an overloaded or low energy state, a discharge message will be transmitted to its clusterhead. The latter consults its members table and chooses the one which has the smallest load and the highest energy capacity. Then it sends a response to the concerned node.

Whenever a new node joins a cluster, or an existing node exits it, the members table is updated. The bases of our load balancing algorithm are as follows:

- Clusterhead nodes maintain their members tables in order to control their members loads. Periodically, each node, member of a cluster, sends a HELLO message and communicates its energy and load values to the clusterhead which updates its table.
- Two thresholds are defined for each node: the maximum load that a node is able to carry out and the minimum energy. When this is reached, the node knows that it is going to die soon, so it decides to transfer its load to another node.
- Periodically, each node checks its load and its energy and compares them with the two thresholds. Two cases are considered:

First case: the two thresholds are not reached, in this case nothing happens and the load balancing algorithm is not invoked.

Second case: if one of the two thresholds is reached (possibly both of them), the node sends a message (DR. Discharge Request) to its clusterhead. This one consults its members table; it chooses the one which has the smallest load and the highest energy capacity.

- If one node is found, the clusterhead sends a positive response (message RESPONSE) to that node indicating the address of the new node that will receive the extra load.
- If several nodes are found, it chooses one arbitrarily.
- If no node is found, the clusterhead diffuses a solicitation message to its different neighbors clusterheads. If it is positive, a receiving node is designated and a quantity of work is transferred to it. If the response is negative, the node is constrained to execute the work locally.

4. Environment of Simulation

The simulation model is carried out by using the Network Simulator (NS-2) [25]. It consists of N (50 to 300) nodes, which move arbitrarily on a surface of (100×100) grid. The nodes can move in all possible directions with a displacement speed which varies between 0 and 10 m/s. Each node generates a number of jobs with a given frequency. The job average size is between 10^8 and 10^9 instructions. Each node calculates its current load by summing up all the jobs assigned to it. Initially, the load threshold value is fixed to $+ 20\%$ of the cluster average load (μ).

To measure our algorithm performance, we have identified the metrics: execution time improvement, energy, balance factor, Load Balancing Threshold and communication cost.

Balance factor: It is difficult to maintain a perfectly balanced system at all times. So, to quantitatively measure the degree of load balance, we introduce a balance factor F defined as follows:

$$F = \frac{n}{\sum_i (load_i - \mu)^2} \quad (1)$$

n: number of nodes in a cluster (its cardinality)
$load_i$: load of a node i

$$\mu = \frac{\sum_i load_i}{n} \quad (2)$$

μ: average load of a cluster.

Load Balancing Threshold value: The threshold is a value that is used to indicate whether a node is heavily or lightly loaded. It is important to determine an appropriate threshold for a good load balancing algorithm. If the threshold is set too low, excessive load balancing will occur, thus causing degradation in performance (as this will result in high communication costs). However, if the threshold is set too high, the load balancing mechanism will not be very effective.

Communication cost: This is the total number of load balancing-related messages sent by a node. It gives a measure of the overhead created for balancing loads, which is also an indication of our algorithm cost in terms of energy.

5. Results Analysis and Interpretation

After simulations, the obtained results are analyzed by studying the impact of the considered parameters on our load balancing algorithm performance.

5.1. Execution Time Improvement

To evaluate the performance of our algorithm in term of execution time improvement, we varied the number of nodes and registered the execution time for different sizes of works (small, average, large).

Figure 1 illustrates the improvement of the execution time according to the nodes number. We notice that the execution time gives better results for a high number of nodes. A clear improvement occurs especially for big sized works. This can be explained by the fact that the higher the number of nodes present in the network, the better the work distribution to other nodes. This improves their execution time.

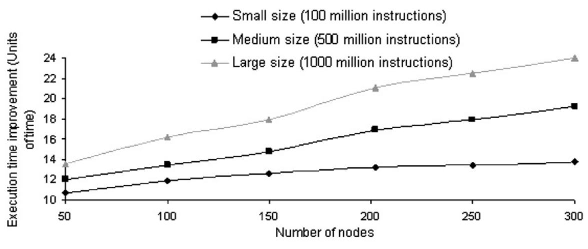

Figure 1. Execution time improvement vs. number of nodes.

For small sizes of works and for different numbers of nodes, the execution time improvement is not really noticeable due to the fact that there is no load imbalance in the network and consequently there is no great need for balancing loads.

5.2. Energy

Figure 2 shows that whatever the network nodes number, we notice that there is a significant reduction of the power consumption after load balancing. In this case, the nodes can survive for long periods of time; this involves a good maintenance of the network stability.

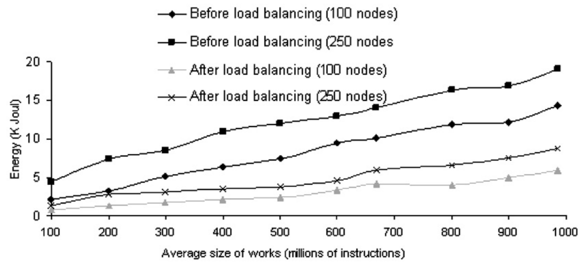

Figure 2. Energy vs. size of works.

5.3. Impact of Mobility on Energy

In this case, we try to see the effect of mobility on the nodes energy, without and with load balancing.

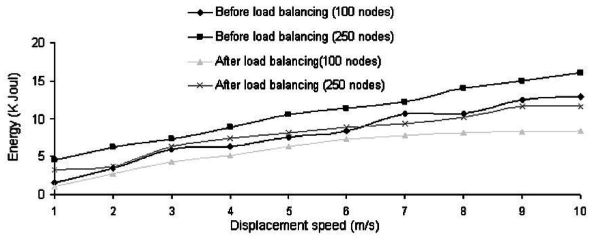

Figure 3. Energy vs. displacement speed.

According to Figure 3, we can say that before balancing, energy consumption varies proportionally to the nodes displacement speed. The more the nodes move, the more they tend to be distant from their clusterheads, thus causing more re-affiliations and more information updates between clusters, which induces higher energy consumption. This situation is not noticed by distributing the load between the different nodes. Indeed, for small speed values, the energy increases slightly. Once a speed value (6) is reached, even though there is a change in the clusters structure, with a given load balance, the energy consumption is stabilized, that extends the lifetime of nodes with weak energy.

5.4. Balance Factor

We remind that the balance factor allows us to measure the load balancing degree between the clusters' members. The higher the value of F, the better the load distribution.

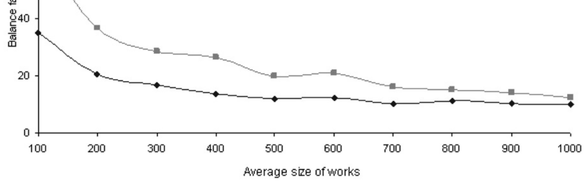

Figure 4. Balance factor after load balancing.

According to Figure 4 we notice that the factor F decreases slightly with the increase of the work size, but it stabilizes later. This explains that our load balancing algorithm guarantees the work distribution uniformity, but it acts favorably for networks having fewer nodes.

5.5. Threshold Value

The threshold is a value used to indicate whether a node is heavily or lightly loaded. It is important to determine appropriate threshold for a good load balancing algorithm. If the threshold is set too low, excessive load balancing will occur, thus causing degradation in performance (as this will result in high communication costs). However, if the threshold is set too high, the load balancing mechanism will not be very effective.

Figure 5 shows that the execution time is reduced as the load threshold value is increased. However, the time execution is increased after 20%. Therefore, having a load balancing algorithm that is too restrictive is inefficient. On the other hand, if the threshold value is high, the load balancing becomes ineffective as overloaded nodes will not be detected until it is too late. This figure leads to an observation that this load balancing mechanism worked best when the threshold value was 20%.

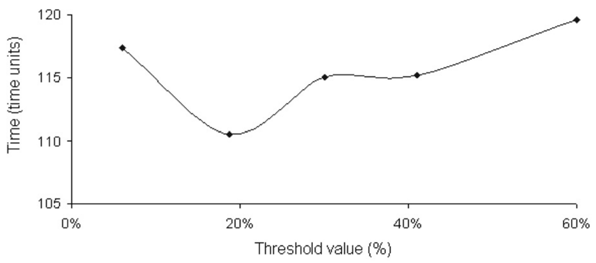

Figure 5. Execution time for different values of load threshold.

5.6. Communication Cost

Figure 6 summarizes our experimental results of the communication costs in each node. It is clear that the network becomes denser while the number of nodes increases. However, experiments show that load balancing algorithm does not cost more messages on each node, even when the network becomes denser. This

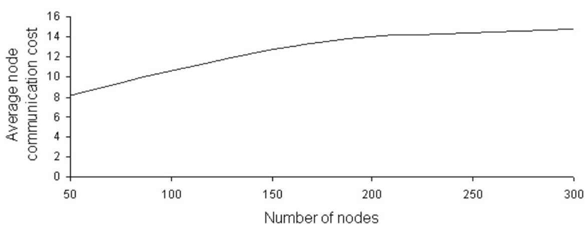

Figure 6. Communication cost vs. number of nodes.

is due to the fact that the used load balancing approach is centralized on the level of each cluster. Indeed, the exchanges are done between the clusterhead and its members. Moreover, simulations results show that the performances of our algorithm are stable when the number of nodes changes.

6. Conclusion and Future Work

Load balancing is an important solution to improve the execution time of tasks and better management of the energy by reducing load imbalances in ad hoc networks. In order to take into account the limitations of these networks in terms of bandwidth and energy, clustering techniques have been suggested. In this paper, we have presented a new load balancing algorithm based on clustering where a subset of nodes 'clusterheads' is elected to coordinate their respective clusters.

The simulation results show a significant improvement of execution time (30%) and a good energy management for a great number of nodes and big sizes of work.

This work is principally based on the clustering. We plan to use various clustering algorithms presented in the literature. The presented algorithm has implicitly assumed that all nodes are cooperative. In some situations some nodes refuse to cooperate in load balancing. It would be interesting to study the impact of selfish nodes on the load balancing algorithm.

References

[1] ALAN, D. A., RAVI, P., Load-Balancing Clusters in Wireless Ad Hoc Networks. *Proceedings of the 3rd IEEE Symposium on Application-specific Systems and Software Engineering Technology (ASSET'00)*, (2000), pp. 25–32.

[2] ANDERSON, T. E., CULLER, D. E., PATTERSON, D. A., THE NOW TEAM, A Puts for NOW (Networks of Workstations). *IEEE Micro*, Vol. 15, No. 1, January 1995, pp. 54–64, `HTTP://now.cs.berkeley.edu/Case/case.ps`.

[3] BAHI, J., COUTURIER, R., VERNIER, F., Synchronous distributed load balancing on dynamic networks. *Journal of Parallel and Distributed Computing*, Vol. 65 No. 11, November 2005, pp. 1397–1405.

[4] BAKER, D. J., EPHREMIDES, A., A distributed algorithm for organizing mobile radio telecommunication networks. In *Proceedings of the 2nd International Conference on Distributed Systems Computer*, April 1981, pp. 476–483.

[5] BASAGNI, S., Distributed clustering for ad hoc networks. *International Symposium on Parallel Architectures, Algorithms and Networks*, Perth, June 1999, pp. 310–315.

[6] BASAGNI, S., Distributed and mobility-adaptive clustering for multi-media Section ort in multi-hop wireless networks. *Proceedings of the Vehicular Technology Conference*, VTC, Flight 2 (1999), pp. 889–893.

[7] BASAGNI, S., MASTROGIOVANNI, M., PANCONESI, A., PETRIOLI, C., Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks*, Vol. 17, No. 4, April 2006, pp. 292–306.

[8] BASU, P., KHAN, N., LITTLE, T. D. C., A Mobility Based Metric for Clustering in Ad hoc Mobile Networks. *Proc. IEEE ICDCS 2001 Workshops on Wireless Networks and Mobile Computing*, Phoenix, AZ, April 2001.

[9] CHATTERJEE, MR., DAS, S. K., TURGUT, D, WCA: A Weighted Clustering Algorithm for mobile Ad Hoc Networks. *Newspaper of Clustering Computing, (Special Mobile Exit on Ad Hoc Networks)*, Vol. 5, No. 2, April 2002, pp. 193–204.

[10] CHEN, B., JAMIESON, K., BALAKRISHNAN, H., MORRIS, R., Span: Energy-efficient year coordination for topology maintenance in Ad Hoc wireless networks. In *Proc. of ACM/IEEE 7 HT International Conf. on Mobile Computing and Networking (MobiCom 2001)*, July 2001.

[11] GAYATHRI, V., SABU, E. SRIKANTHAN, T., Size-restricted cluster formation and cluster maintenance technique for mobile Ad Hoc Networks. *International Journal of Network Management*, Vol. 17, Issue 2, March 2007, pp. 171–194.

[12] GERLA, M., TSAI J. T. C., Multicluster, mobile, multimedia radio network, wireless networks, Vol. 1 (3), 1995, pp. 255–265.

[13] HASSANEIN, H., ZHOU, A., Routing with Load Balancing in Ad Hoc Wireless Networks. *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Rome, Italy, July 2001, pp. 89–96.

[14] HO, C. K., EWE, HONG, A hybrid ant colony optimization approach (hACO) for constructing load-balanced clusters. Evolutionary Computation, 2005. *The 2005 IEEE Congress*, Vol. 3, Sept. 2005, pp. 2010–2017.

[15] JANE, Y. YU, PETER, H. J., A Survey of Clustering Schemes for mobile Ad Hoc networks. *IEEE Communications Survey and Tutorials*, 2005.

[16] JANE, Y. YU, PETER, H. J., An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs). *Computer Communications*, Vol. 30, Issue 1 (December 2006), pp. 5–16.

[17] JEANNOT, E., VERNIER, F., A Practical Approach of Diffusion Load Balancing Algorithms. *Euro-Par 2006*, pp. 211–221.

[18] JOHANSSON, T., CARR-MOTYCKOVA, L., On Clustering in Ad Hoc Networks. *Division of Computer Science and Networking*, Lulea University of Technology, August, 2003.

[19] KWANG, M. S., WENG, H. S., Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 33, No. 5, September 2003.

[20] OZAN K. TONGUZ, EVSEN YANMAZ, On the Theory of Dynamic Load Balancing. *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, Vol. 7, pp. 3626–3630.

[21] RONG, Z., ROBIN, K., On-demand Power Management for Ad Hoc Networks. *IEEE INFOCOM 2003*.

[22] SANON, C., KOMWUT, W., SUWAN, R., Load Balancing for Zone Routing Protocol to Support QoS in Ad Hoc Network. `citeseer.ist.psu.edu/chimmanee02load.html`.

[23] TAKAHASHI, S., HAKODA, J., UEHARA, H., YOKOYAMA, M., A Load Balanced Routing Scheme for Mobile Ad Hoc Networks. *International Symposium on Information Theory and its Applications*, ISITA2004, Parma, Italy, October 2004.

[24] TUGRUT, D. K., DAS, S. K., ELMASRI, R., Balancing loads in Ad Hoc Mobile Networks. *ICT 2003. 10th International Conference on Telecommunications, 2003*, Vol. 1, pp. 490–495.

[25] UCB/LBNL/VINT, Network simulator-NS (version2). `http://www-mash.cs.berkeley.edu/ns/january`.

[26] WILLEBEEK-LEMAIR, M., REEVES, A. P., Strategies for Dynamic Load Balancing on Highly Parallel Computers. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 4 (9), 1993, pp. 979–993.

[27] YASHAR, G., ABTIN, K., Load Balancing in Ad Hoc Networks: Single-path Routing vs. Multipath Routing. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, Vol. 2, 2004, pp. 1120–1125.

[28] ZARIFZADEH, S., YAZDANI, N., KHANMIRZA, H., A routing framework for load balancing of bandwidth sensitive traffic in differentiated service networks. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 51, Issue 4 (March 2007), pp. 1183–1204.

[29] ZHOU, S., A Trace-driven Simulation Study of Dynamic Load Balancing. *IEEE Transactions on Software Engineering*, Vol. 14(9), September 1988, pp. 1327–1341.

[30] ZHOU, S., WANG, J., ZHENG, X., DELISLE, P., Utopia: In Load Sharing System for Broad, Heterogeneous Distributed Systems Computer. *report n_CSRI257, Computer Systems Research Institute*, University of Toronto, Toronto, Canada, April, 1992, `ftp://ftp.cs.toronto.edu/pub/reports/csri/257/257.ps.Z`.

Received: January, 2008
Accepted: December, 2008

Contact address:
Rachida Aoudjit
Département d'informatique
Université de Tizi-Ouzou
Algérie
e-mail: `aoudjit_rachida@yahoo.com`

RACHIDA AOUDJIT is an Assistant Professor of electrical engineering at the Faculty of Electrical Engineering, University of Tizi-Ouzou. She is also a research member at the LARI Laboratory of the Computer Science Department. Her areas of interest include mobile networks, computer architecture and embedded systems.

MUSTAPHA LALAM received the Master's degree in Computer Architecture from the High School of Computer Science, Algiers, Algeria, in 1980 and the Ph.D. degree in Computer Science from University of Toulouse, France, in 1990. He joined the University of Tizi Ouzou, Algeria in 1993, where he is now Professor in the Computer Science Department at the University of Tizi Ouzou. He has been involved in research and Development of Computer Architecture, Distributed Systems and Mobility management for Wireless Mobile Computing and Communications.

ABDELAZIZ M'ZOUGHI is a Professor at Computer Science Institute, University of Toulouse, France. His areas of interest include mobile networks, computer architecture and embedded systems.

MALIKA BELKADI is an Assistant Professor of electrical engineering at the Faculty of Electrical Engineering, University of Tizi-Ouzou. She is also a research member at the LARI Laboratory of Computer Science Department. Her areas of interest include mobile networks, and embedded systems.

MEHAMMED DAOUI is an Assistant Professor of electrical engineering at the Faculty of Electrical Engineering, University of Tizi-Ouzou. He is also a research member at the LARI Laboratory of Computer Science Department. His areas of interest include mobile networks, and embedded systems.