Improved Topology Control Algorithm for MANETs

Hatem Hamad*

**Department of Electrical and Computer Engineering*
The Islamic University of Gaza, Palestine
hhamad@mail.iugaza.edu

Abstract: A mobile wireless ad hoc network is formed dynamically without a need for a pre-existing infrastructure. Frequent Topology changes leads to more processing and hence more power consumption. Reducing power consumption during node's life time is a challenging task. Adaptive Self-Configuring sEnsor Networks Topologies (ASCENT) is a topology control technique for reducing the power consumption during the node lifetime. In ASCENT, each node assesses its connectivity and adapts its participation in the multi-hop network topology based on the operating region. In this paper, I study some of the problems in ASCENT algorithm and propose a modified state diagram, which adaptively adjusts the states of individual nodes, to reduce redundancy. This helps to achieve the optimum number of Active nodes in the network. I implement the modified state diagram, and simulation results highlight that the improved ASCENT state diagram is able to achieve better performance than the original ASCENT algorithm.

Keywords: Topology Control, ad hoc networks, Power Consumption

1. INTRODUCTION

Topology control is an important way to reduce the consumption of power, that's why many algorithms (Chen et al. 2002, Deb et al. 2005, Hu. 1993, Ahmed et al. 2005, Wattenhofer et al. 2005, Ramanathan et al. 2000) were developed to get the best performance of the mobile ad hoc networks (MANETs). This paper discusses the node scheduling scheme by improving the performance of ASCENT algorithm (Cerpa et al. 2004). In general, coordination between nodes reduces the redundancy in the high density networks. This helps in extending the overall system lifetime. Also having too many nodes deployed in a wide range area is very difficult to manage and configure manually, design-time pre-configuration is precluded because of the environmental dynamics. That’s why self-configuration is used to achieve the desired topology control.

ASCENT algorithm depends on the concept of node scheduling to achieve topology control (Cerpa et al. 2004, Deb et al. 2005). It has four states to represent node's active/inactive state. Active states are:

- *Active:* Forwards data and routes packets.
- *Test:* Sends neighbour announcement message, monitors network for neighbours and data loss rates and forwards data and routes packets.

Inactive states are:

- *Passive:* Monitors network for neighbours and data loss rates, also periodically checks if necessary to become active.
- *Sleep:* Turns radio off and goes to sleep.

ASCENT consists of several phases. When a node first initializes, it enters into a listening-only phase called neighbour discovery phase, where each node obtains an estimate of the number of neighbours actively transmitting messages based on local measurements. Upon completion of this phase, nodes enter into the join decision phase, where they decide whether to join the multi-hop diffusion sensor network. During this phase, a node may temporarily join the network for a certain period of time to test whether it contributes to improved connectivity.

If a node decides to join the network for a longer time, it enters into an active phase and starts sending routing control and data messages. If a node decides not to join the network, it enters into the adaptive phase, where it turns itself off for a period of time, or reduces its transmission range (Cerpa et al. 2004).

2. STATE OF THE ART

My work has been informed and influenced by a variety of other research efforts. There has been a great deal of work in the area of topology control, mostly using theoretical analysis or simulation, and involving MAC (IEEE Standards 1994) and power control mechanisms (Agarwal et al. 2001, Chen et al. 2002).

There have been several important theoretical evaluations of topology control. Most of this work focuses on node scheduling algorithms. They are used to control topology by reducing number of routers especially in dense network. There are a lot of works related to this field.

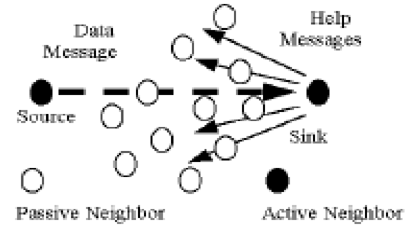
Alberto Cerpa and Deborah Estrin proposed self-configuring algorithm (Cerpa et al. 2004) that uses node scheduling concept. This algorithm is the basic of my work. it states that not all nodes are supposed to work as routers in the dense network. Chen proposes a localized algorithm called SPAN for node scheduling (Chen et al. 2002). Using simulations they proved that the node-scheduled topology does not suffer too much in terms of latency and capacity of the network while reducing redundant power consumption. However none of the papers consider the impact of node scheduling on the overhead of reliable transmissions for ad hoc networks. Another important work I referred to during my work was the analysis of algorithms for distributed construction of a connected dominating set (CDS) of the corresponding unit-disk graph and the routing strategies using the CDS backbone (Wan. et al. 2002, Bharghavan et al. 1997).

3. ASCENT ALGORITHM

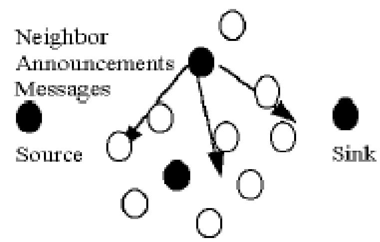
ASCENT adaptively elects "Active" nodes from all nodes in the network. Active nodes stay awake all the time and perform multi-hop packet routing, while the rest of the nodes remain "passive" and periodically check if they should become active. Initially, only some nodes are active. The other nodes remain passively listening to packets but not transmitting. This situation is depicted in Fig. 1(a). The source starts transmitting data packets toward the sink. Because the sink is at the limit of radio range, it gets very high packet loss from the source. This situation is called a communication hole; the receiver gets high packet loss due to poor connectivity with the sender. The sink then starts sending help messages to signal neighbours that are in listen-only mode, also called passive neighbours, to join the network. When a neighbour receives a help message, it may decide to join the network. This situation is illustrated in Fig.1(b). When a node joins the network it starts transmitting and receiving packets, i.e. it becomes an active neighbour. As soon as a node decides to join the network, it signals the existence of a new active neighbour to other passive neighbours by sending a neighbour announcement message. This situation continues until the number of active nodes stabilizes on a certain value and the cycle stops (see Fig. 1(c)). When the process completes, the group of newly active neighbours that have joined the network makes the delivery of data from source to sink more reliable. The process will re-start when some future network event (e.g. node failure) or environmental effect (e.g. new obstacle) causes packet loss again (Cerpa et al. 2004).

3.1 ASCENT state transitions

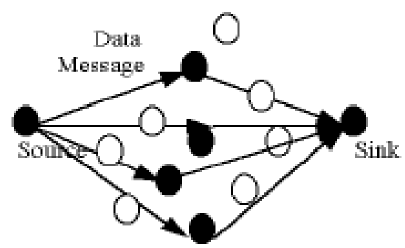
In ASCENT, nodes are in one of four states: sleep, passive, test, and active. Fig. 2 shows a state transition diagram. Initially, a random timer turns on the nodes to avoid synchronization. When a node starts, it initializes in the test state.



(a) Communication Hole



(b) Transition



(c) Final State

Figure 1: Network self-configuration

Nodes in the test state exchange data and routing control messages. In addition, when a node enters the test state, it sets up a timer Tt, and sends neighbour announcement messages. When Tt expires, the node enters the active state.If before Tt expires the number of active neighbours is above the neighbour threshold (NT), or if the average data loss rate (DL) is higher than the average loss before entering in the test state, then the node moves into the passive state. If multiple nodes make a transition to the test state, then I use the node ID in the announcement message as a tie breaking mechanism (higher IDs win). The intuition behind the test state is to probe the network to see if the addition of a new node may actually improve connectivity.

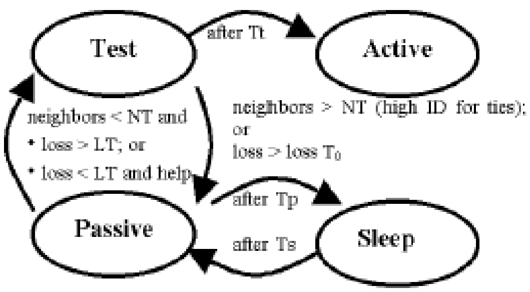


Figure 2: ASCENT state transitions

When a node enters the passive state, it sets up a timer Tp and sends new passive node announcement messages. This information is used by active nodes to make an estimate of the total density of nodes in the neighbourhood. Active nodes transmit this density estimate to any new passive node in the neighbourhood. When Tp expires, the node enters the sleep state. If before Tp expires the number of neighbours is below NT, and either the DL is higher than the loss threshold (LT) or DL is below the loss threshold but the node received a help message from an active neighbour, it makes a transition to the test state. While in passive state nodes have their radio on, and are able to overhear all packets transmitted by their active neighbours (even if the packets are not addressed to the passive node, since the radio is in promiscuous mode). No routing or data packets are forwarded in this state, since this is a listen-only state. The intuition behind the passive state is to gather information regarding the state of the network without causing interference with the other nodes. Nodes in the passive and test states continuously update the number of active neighbours and data loss rate values. Energy is still consumed in the passive state, since the radio is still on when not receiving packets. A node that enters the sleep state turns the radio off, sets a timer Ts and goes to sleep. When Ts expires, the node moves into passive state. Finally, a node in active state continues forwarding data and routing packets until it runs out of energy. If the data loss rate is greater than LT, the active node sends help messages (Cerpa et al. 2004).

4. PROBLEMS

Active nodes are responsible for routing, they construct the backbone of the network and form its topology (Santi 2005). Having too many Active nodes consumes a lot of unnecessary energy.

The original state diagram has no returning path from Active state, as shown in Fig. 2. There are many cases and scenarios that may cause this problem. For example, if the Active node changes its position due to the mobility of the network. The new position may have no routing tasks which means the node will be Active with no tasks (Idle node). There is no need for this node to be Active, so it is better to return to Passive again.

Another problem can arise if the new position has enough Active nodes, which means they will exceed Neighbour Threshold (NT) (Cerpa et al. 2004). Too many Active nodes cause a lot of overhead in the network. When a node sends help message due to packet loss, there might be several responded nodes. This may cause to extra numbers of Active nodes eventually. A similar scenario happens when an Active node is lost because of the mobility or node's failure. The gap will be detected and a help message will be sent to find a replacement. Then, several nodes will replace the failed node.

My solution suggests that it is better to drain an area of nodes more slowly than to drain a node completely. I solved this problem by adding Active withdrawal state to the original state diagram as discussed in the next section.

5. SOLUTION

As stated in the previous section, the original state diagram doesn't fix the problem of the unnecessary number of nodes entering Active states, it has no mechanism to check the need for these Active nodes. Also the node that enters the Active state remains Active until its power is drained. To solve the previous problems a new modified state transition diagram is presented which will handle the additional issues provided previously. A new state was added to the original diagram called "Active Test". It is responsible for testing the necessity of the node to stay in the Active state or not, therefore reducing the power consumption in the network. The purpose for this modification is to reduce the unnecessary number of Active nodes. To achieve this, the node will move to *ActiveTest* state if it has no routing tasks, Fig 3. This should be done with regard to the network-wide information, particularly the number of Active neighbours, in order to guarantee the connectivity between nodes in the network and to ensure that only the unneeded nodes returns to passive. During *ActiveTest* state, the node has no routing tasks and ready to go to Passive state if possible. Being in Active state consumes energy in sending Neighbour Announcement messages. To maintain network's connectivity the node will check number of Active neighbours to insure that it will not affect the connectivity.

The transition from Active to ActiveTest happens when the node has no routing tasks for $\triangle t$, I assumed $\triangle t$ = Tt* in the original algorithm. This is an indication that the node might be useless for the network; it should go through several tests as described next to check if the node is needed to stay in Active state or not. This is done in the ActiveTest state. During ActiveTest state, the node will check the number of its Active Neighbors. Having too many Active nodes makes the network very crowded, the interference is increased due to the heavy message exchange between nodes. This will affect the performance of the network (Gang et al. 2006). If number of Active Neighbors > Neighbors Threshold (NT), the node will return to Passive state, Fig. 3.

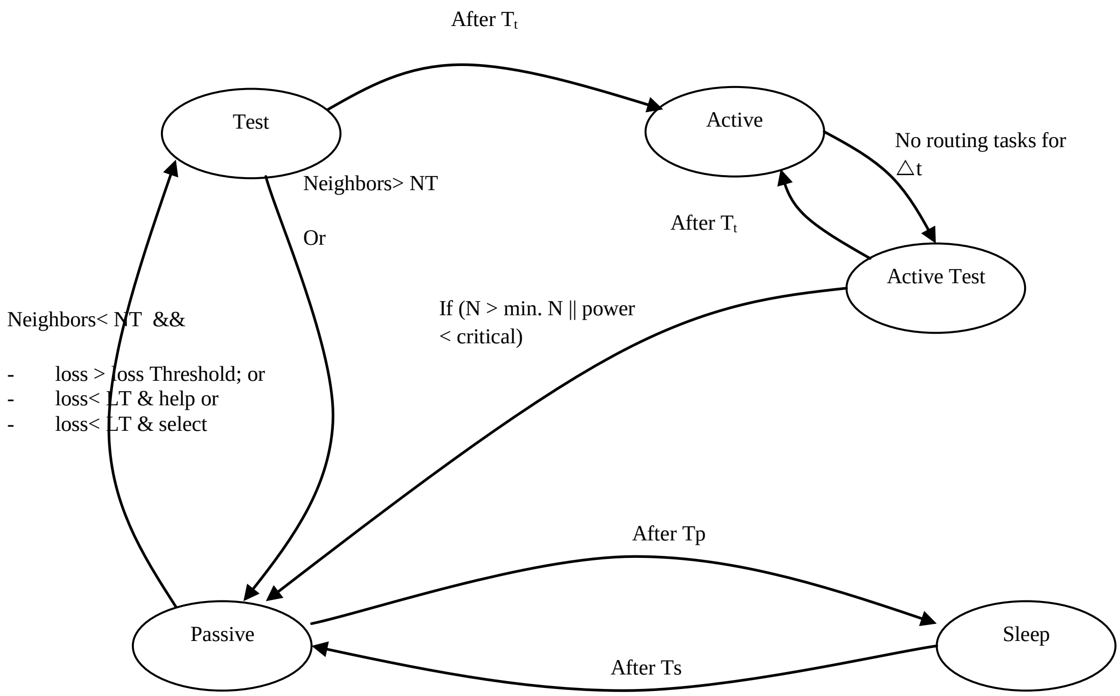


Figure 3: Modified state diagram

Such partition of the network is very crowded and has too many Active nodes. Redundancy between Active nodes can be reduced by removing the unnecessary ones with no routing tasks in order to maintain the routing paths. This situation happens due to the mobility in the network.

Another important measure is the power level of the node. If a node is in critical power which I assume after experiments to be 5%, it should not work as a router for other nodes consuming its remaining power. This could lead to a sudden death of that node. One way to solve this issue is to return to Passive state after selecting one of its Passive neighbours to go to Active. A unicast message is sent by that node to the selected Passive node. The selection criteria is the power level of that node, the neighbour Passive node with the highest power level will receive the message "Select Message" and move to Test state as described in Fig. 3.

After Tt*, the node will return to Active if non of the previous conditions were achieved. This indicates that the node is still needed in the network and has a role in the topology formation and future routing tasks, Fig. 3. In general, this modified state diagram solves the problems of achieving the optimum number of Active nodes by reducing their numbers and withdrawing the unneeded to the Passive state.

The resulted redundancy will be reduced as well as the power consumption of the useless Active nodes. Another advantage is reducing the sudden gap in the network which happens because of the sudden death of the node after its power is drained.

6. SIMULATION AND RESULTS

I used the JiST-SWANS simulator (Barr 2006a,b) to simulate the proposed solution. My simulation compares the performance of the original ASCENT and the modified algorithm. The results show the changes of performance in the Network life time and Delivery ratio (network connectivity). My evaluation is based on the simulation of different node densities in the field. For measuring connectivity I used 20 node in different field range 100x100 to 3000x3000 Km^2 with transmission range of each node equals to 625 meters, two-ray ground propagation channel is assumed with a data rate of 1 Mbps. For measuring network life I took multiple values during simulation time at 0 to 1000 seconds with 50 second increment in 1500x1500 meters field, repeating the results for 20, 40 and 80 nodes. The data traffic simulated is constant bit rate (CBR) traffic (Perkins 2001). 50% of nodes, CBR sources, generate ten 128-byte data packets every (20-25) second. Random waypoint mobility model was used in my experiments with a maximum node speed of 2 m/s and a pause time 500 ms. With this approach, a node travels towards a randomly selected destination in the network. After the node arrives at the destination, it pauses for the predetermined period of time and travels towards another randomly selected destination. Simulation time is 2000 seconds and each simulation scenario is repeated several times to obtain steady-state performance metrics. In order to provide a fault situation, I used Uniform packet loss model that drops packets at certain probability. I used AODV as a routing protocol between the nodes (Perkins et al. 2000)

6.1 Average Network life time

This is a measure of the network life time. The value is measured by summing the power percent of each node in the network divided by the number of nodes.

Avg. Network Life time = $((P1 + P2 + ... + Pi)/i)$ (1)

where *i* is the number of nodes in the network.

The simulation was implemented in several densities, 40 and 80 nodes in 1500x1500 meters (18 and 36 nodes/Km2). The results are as shown in the figures below, Fig. 4, Fig. 5. In these figures, I can see clearly that improved algorithm extends the networks lifetime comparing to the actual algorithm. This is very useful and cost-effective for different network applications. I can observe more improvement in dense network (80 node). The improvement happened because of the Active withdrawal and optimum number of Active nodes, draining power in the improved algorithm comes from an area of nodes which is more slow and efficient than draining one node completely.

6.2 Success Delivery ratio

This parameter indicates the effect of my algorithm on network's connectivity. The previous work causes less number of nodes to be active, which might decrease the network connectivity. The success delivery ratio (SDR) is computed by dividing the number of received packets over the number of sent packets.

SDR = $\frac{(number\_of\_received\_packets)}{(number\_of\_sent\_packets)} x100\%$ (2)

The simulation was implemented in several densities by increasing the field range. I repeated the simulation first on static network then on mobile network with Random Way point model explained previously. In the static mobility, I observe no significant change in the success Delivery ratio compared to my modified algorithm, Fig. 6. This is because there is no often error case and path broken in the static network, which reduces the chances of my algorithm to take effect on the overall performance. After all I can see a little improvement in the high density network. In the dynamic mobility I can observe the increasing of the performance on the success delivery ratio especially at low density, Fig. 7. This is because my algorithm maintains a minimum number of Active nodes and in the case of Active node's failure, the node selects a replacement which will help in improving the connectivity.

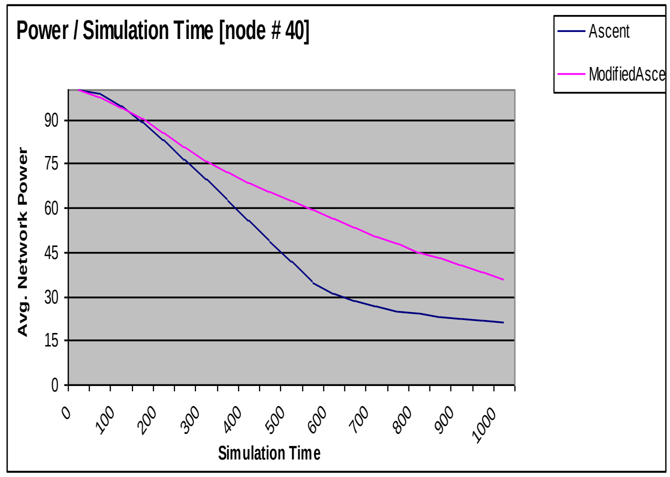


Figure 4: Network life time at 40 nodes

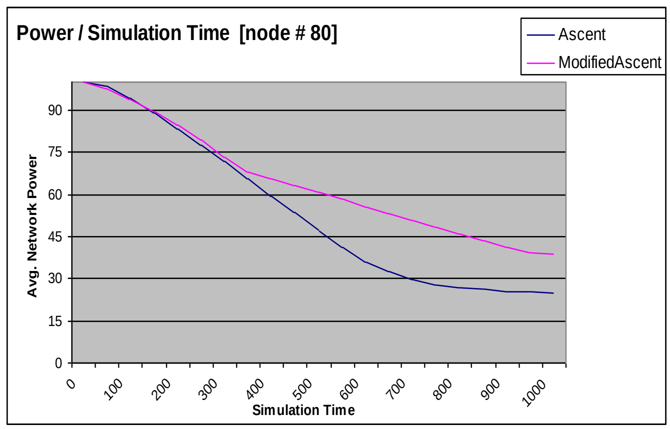


Figure 5: Network life time at 80 nodes

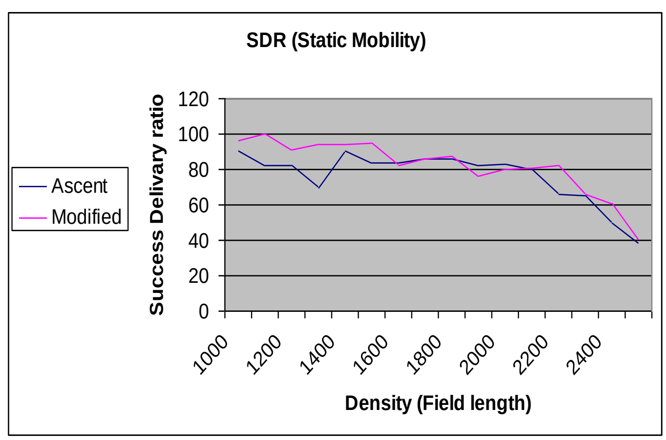


Figure 6: SDR at static Mobility

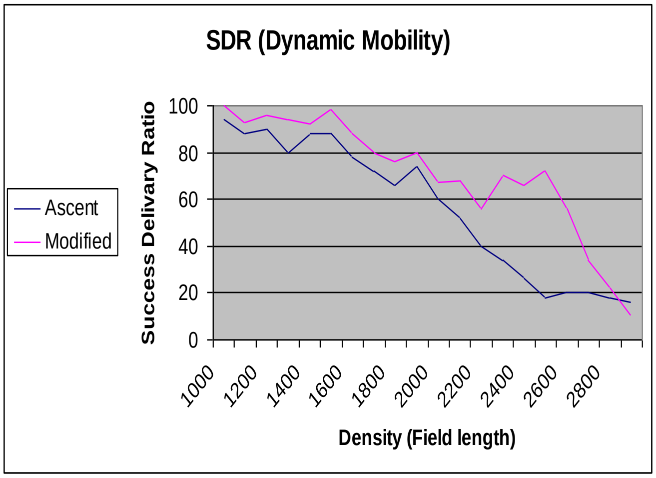


Figure 7: SDR at Dynamic Mobility

7. CONCLUSION

In this paper I presented a solution to some problems in ASCENT topology control algorithm. The solution is a modified state diagram that provides the following advantages to the original algorithm: (a) achieves optimum number of Active nodes by adding "Active withdrawal" stage, (b) reduces the redundancy resulted by a crowded network and (c) reduces the gap in the network which is resulted by the sudden death of the node after its power is drained.

I hope in the future work to solve other open issues related to ASCENT Algorithm, such as the problem of Neighbor Threshold value. In the original ASCENT the value is fixed and depends on the application. In general, it is better to have a dynamic value of threshold for different environments and different types of nodes (in case of heterogeneous network). Another open issue is that broadcasting help message will cause nodes to go from passive to test states needlessly in areas where help is not needed, increasing collisions in that area temporarily. My solution handles this issue after detecting it, but this could be improved by not allowing these nodes to enter Active state from the start.

ACKNOWLEDGEMENT

The author would like to thank the research assistant Eng. Mohammed H. Alafifi for his assist in doing this work.

REFERENCES

[1] Agarwal S., Krishnamurthy S., Katz R., Dao S. (2001). Distributed Power Control in Ad-hoc Wireless Networks. *PIMRC01*.

[2] Ahmed A., Homayoun B. (2005). Topology discovery for network fault management using mobile agents in Ad-Hoc networks. *IEEE 0-7803-9305-8/05*.

[3] Barr R. (2004). JiST– Java in Simulation Time User Guide, *Department of Computer Science, Cornell University, Ithaca, NY 14853*.

[4] Barr R. (2004). JiST SWANS– Scalable Wireless Ad hoc Network Simulator User Guide, *Department of Computer Science, Cornell University, Ithaca, NY 14853*.

[5] Bharghavan V. and Das B. (1997). Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets. *International Conference on Communications' 97, Montreal, Canada*.

[6] Cerpa A., Estrin D. (2004). ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies. *IEEE 1536-1233/04*.

[7] Chen B., Jamieson K., Balakrishnan H., Morris R. (2002). Span: An Energy Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. *ACM*

[8] Chen B., Jamieson K., Balakrishnan H., Morris R. (2002) Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks. *IEEE 0-7803-7476-2*.

[9] Deb. B., Nath B. (2005). On the Node-Scheduling Approach to Topology Control In Ad Hoc Networks. *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. International Symposium on Mobile Ad Hoc Networking & Computing*.

[10] Gang Z., Stankovic J., Son S. (2006), Crowded Spectrum in Wireless Sensor Networks, *unpublished*.

[11] Hu L. (1993). Topology control for multihop packet radio networks. *IEEE Trans. Communications*, **vol. 41, no. 10**.

[12] IEEE Standards Department (1994). Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Standard 802.11_1997.

[13] Perkins C. (2001). *Ad Hoc Networking*, Addison-Wesley.

[14] Perkins C., Royer E., and Das S. (2000). Ad Hoc On Demand Distance Vector (AODV) Routing. *IETF Internet Draft, draft-ietf-manet-aodv-05.txt*.

[15] Ramanathan R., Rosales-Hain R. (2000). Topology control of multihop wireless networks using transmit power adjustment. *IEEE INFOCOM*.

[16] Santi P. (2005). Topology Control In Wireless Ad Hoc And Sensor Networks, *John Wiley & Sons*.

[17] Wattenhofer. R., Li L., Bahl P., Wang Y. (2005). Distributed topology control for wireless multihop ad-hoc network. *IEEE INFOCOM*.