International Journal of Soft Computing 4 (5): 185-196, 2009
ISSN: 1816-9503

Evaluation of Network Life Time of Wireless Sensor Networks Using Hierarchical Energy Tree Based Routing Algorithm with Exponential Congestion Control

[1]M. Kalpana and [2]T. Purusothaman
[1]Government College of Technology, Anna University, Coimbatore 641013, Chennai, India
[2]Department of CSE, Government College of Technology, Coimbatore 641013, India

Abstract: This study evaluates the Network Life Time (NLT) of Wireless Sensor Networks. The Wireless Sensor Network uses a new Routing Algorithm proposed by the researchers. This algorithm is called Hierarchical Energy Tree based Routing Algorithm (HETRA). The aim is to have a routing algorithm with proper congestion control. An exponential congestion control is proposed. This combination is designed to improve the energy efficiency of the WSN and there by increase the NLT. The routing algorithm uses a Hierarchical Energy Tree (HET), which is constructed using the available energy in each node of the network. This HET is used to establish either a bidirectional link or unidirectional link with each node based on the remaining energy. This Algorithm also attempts to solve the data consistency problem due to dynamic nature of the sensor networks. With the proposed algorithm specific distribution patterns of sensor nodes are also assumed in this study for finding out the Routing Protocol along with the Congestion Control Algorithm that could improve the NLT of Wireless Sensor Networks. The analysis is performed by the simulation of wireless sensor networks with HETRA and the three routing protocols used in mobile ad hoc networks-AODV, DSDV and DSR. The Protocol assumed in the Transmission layer is TCP. Five variants of TCP congestion control are considered. They are TCP/Exp, MIMD-Poly, PIPD-Poly, TCP/Tahoe (simply called as TCP) and TCP/Reno. The Network is simulated using ns2. Five different node distribution patterns are assumed. Their performances are compared based on NLT, energy efficiency and a new parameter called Data Packets-ANLT (DANLT) product. The results show that if the sensor nodes are distributed as a CROSS pattern implemented with HETRA Routing protocol and TCP/Exp congestion control algorithm increases the DANLT product.

Key words: Wireless sensor networks, network life time, node distribution, ns2, HETRA, AODV, DSDV, DSR, TCP, AIMD, congestion control algorithms, MIMD-Poly, PIPD-Poly, TCP/Exp, data packets-ANLT product

INTRODUCTION

A number of applications today necessitated the use of Wireless sensor networks and their deployments will have significant impact on both scientific adventures and the day to day life (Cheng *et al*., 2008; Dhillon and Chakrabarty, 2003). Sensor nodes with limited resources in terms of sensing range, placement or distribution patterns, computational power, memory, battery power and transmission capacities comprise Wireless Sensor Networks (WSN). Among these basic issues of WSN, the energy consumption and Network Life Time (NLT) are the major concerns (Wan *et al*., 2002; Akyildiz *et al*., 2002).

This problem is addressed in this study for a class of sensor networks. The battery power carried by a sensor node is limited. For many applications, the desired lifetime of a sensor network is of the order of a few years (Wan *et al*., 2002). It may be infeasible or undesirable to change batteries in sensor nodes once a wireless sensor network is deployed. Thus, it is critical and challenging to design long-lived sensor networks with the energy constraints (Pan *et al*., 2003; Howard *et al*., 2001). For increasing the NLT the network should be implemented with a suitable Routing Algorithm, Congestion control Algorithm and Node distribution patterns. In this study, we study the deployment of sensor networks that will improve the NLT with minimum number of nodes (Winfield, 2000; Alan Mainwaring *et al*., 2002; Wu *et al*., 2000; Chandrasekaran and Banu, 2006a). There are applications where the placement of sensor nodes can be controlled instead of randomly deployed. For example, sensors used to monitor bridges are usually precisely placed. In addition, hierarchical structure is likely to be needed in large-scale sensor networks (Bansal *et al*., 2002; Clare *et al*., 1999).

Corresponding Author: M. Kalpana, Government College of Technology, Anna University, Coimbatore 641013, Chennai, India

Five different distribution patterns of sensor nodes are analyzed in this study. The efficiency of these patterns is evaluated assuming four routing protocols used for the ad hoc networks. They are AODV, TORA, DSDV and the new proposed routing algorithm HETRA (Hou *et al*., 2003). Wireless networks which get connected to Internet, has to use the TCP in their protocol stack (http://www.isi.edu/nsnam/ns; Dunkels *et al*., 2004). Congestion in TCP networks is controlled by either adjusting the window size or by adjusting the sending rate. There are different window based congestion control algorithms available (Hou *et al*., 2003; Dunkels *et al*., 2004). This study simulated the implementations of the following TCP congestion control algorithms: TCP/Tahoe and TCP/Reno. Two congestion control algorithms developed and implemented in ns2 MIMD-Poly and PIPD-Poly (Chandrasekaran and Banu, 2006b; Chandrasekaran *et al*., 2006a, b; Kalpana and Purusothaman, 2009) are also used for comparison. TCP/Exp, the congestion control algorithm proposed by the researchers (Kalpana and Purusothaman, 2009) is also considered. These five congestion control algorithms with the four Routing algorithms are simulated using the five node placement patterns.

Data consistency problem means data retrieved from a location in sensor networks should be consistent with data sent to the same location. This is a great challenge because of the dynamic nature of sensor networks.

We propose an energy driven routing protocol called Hierarchical Energy Tree based Routing Algorithm (HETRA), which routes a packet to the home node of its target location more efficiently. This protocol makes use of the energy remaining in the sensor nodes as a routing metric. HETRA not only works well under the situation where non-uniform transmission ranges exist but also is able to take advantage of the unidirectional links in sensor networks, which would be very useful in practical sensor networks.

The Algorithm makes use of a Hierarchical Energy Tree with all the nodes of the network in one of the levels of the tree. The tree is assumed to be a three level tree. Using this, routing decisions and the unidirectional or bidirectional characteristics of the links are fixed. As a result, unnecessary communication between neighbors is avoided and the valuable energy is saved. A node broadcasts a one-hop beacon-request packet to all neighbors requesting energy information when necessary (Clare *et al*., 1999). Using this information the HET is formed for each node with that node as the root node. The node specifies the type of beacon packets its neighbors use to reply.

The routing decision for packets is made using different neighbor information using the energy metric and link information based on HET. Packets are forwarded greedily, whenever possible based on both bidirectional and unidirectional neighbor information. When a packet reach a dead end with no closer neighbor, the packet switches to perimeter forwarding mode and uses right-hand rule to take tours of enclosed cycles in a planarized network graph. This planarized graph is constructed only based on information of bidirectional neighbors.

Wireless sensor network is simulated using ns2 (Gavin and Vaidya, 2002). This simulator is used by most of the network researchers.

ENERGY AWARE ROUTING USING HETRA

The Hierarchical Energy Tree based Routing Algorithm (HETRA) extends the set of parameters used for taking a routing decision with the inclusion of the energy remaining in the nodes. Energy is an absolute value measured in watt-hour. In addition to the energy used as routing metric, they are also used to place the nodes in a hierarchical tree of three levels of nodes. This is used to categorize the links connecting these nodes as unidirectional or bidirectional. Nodes include such parameters into periodic location update packets, which in our implementation corresponds to an overhead of two octets added to the header.

Energy aware routing: As soon as position update is available, every node is able to instantaneously update location coordinates of its neighbors without the need for communication. To this aim, each node is provided with a high resolution timer upon expiration of which it traverses the neighbor node table adjusting neighbors positions accounting for the range they moved since the last position update beacon was received from them. In addition, the destination coordinates carried inside every transmitted data packet are adjusted accordingly.

Assuming B and Q intervals implemented with a 1 sec. resolution timers, a node is not likely to change much its position between beacon updates. As a result, the adjusted positioning information of the neighbors will be highly accurate.

Furthermore, in order to ensure up-to-date coordinates registered in the location lookup service, the destination node computes the positioning error by checking destination coordinates in every data packet inserted by the source node. In case, the difference of the received (destination) coordinates with real ones is above a predefined threshold, the destination proactively issues position update message to the location lookup service. This threshold can be set as a fraction of the transmission range or represent a predefined parameter.

Routing metric: In order to increase path robustness with efficient energy utilization, the proposed routing metric favors relatively stable paths. To this aim, we define a metric which depends on: distance from the destination and energy remaining in each node as follows:

$$R\ (d,p) = \alpha_d \times f\ (d) +\_\alpha_p \times g\ (p) \quad (1)$$

Where:

α_d and α_p = Parameters with respect to the distance and energy of each node
f(d), g(p) = The functions of distance and energy related to each node

The distance component chooses forwarding nodes located closer to the destination, but considering that the next hop must be within the transmission range and preferably not to close to its perimeter (therefore, excluding nodes closer than a tolerance factor):

$$f_i(d) = e^{(-(d-l))} \quad (2)$$

Where:

$l = d_i$ = Transmission range + tolerance
d_i = The distance of current node i from destination and tolerance is a parameter which will depend on the scenario: the higher the mobility of nodes the higher is its value

Energy that remains in each node after the transmission of a packet is the other metric used for routing. Higher the remaining energy, more probable is the participation of the node in forwarding the packet. The energy measure is given by:

$$g_i(p) = p_i\ (1 - k \times e^{-n_i})) \quad (3)$$

Where:

p_i = The initial energy in node i
n_i = The number of packets transmitted from node I
k = Constant based on the level of hierarchy of energy tree

HET: Hierarchical energy tree is constructed by placing each node of the sensor network in one of the three levels. For each node based on the energy update information received from neighbors, a level is fixed in the HET. One HET is constructed for the entire network and a copy this is sent to each node during update operation. The destination node is kept as the root node (level 0) of the tree. All the remaining nodes are divided into two categories based on their remaining energy. A threshold is used to evaluate the energy level in HET. If the node has energy above the threshold, they are placed in level 1. The remaining nodes are placed in level 2 as shown in Fig. 1.

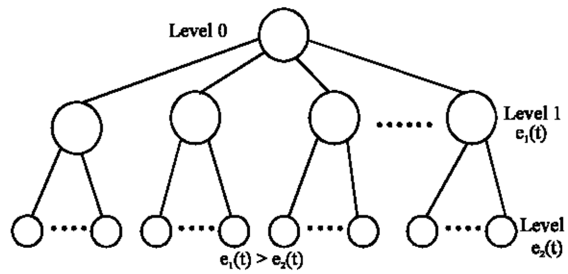

Fig. 1: Hierarchical energy tree

Nodes in level 1 are assumed with bidirectional links and nodes in level 2 are assumed with unidirectional links. The level 2 nodes are attached to the nodes at level 1 if they are the neighbors. So, HET after construction will contain the information of the energy level, link characteristics and neighborhood details. The HET so constructed is used along with the routing metric to determine the routing.

Using HET: HETRA is a reactive energy-driven routing protocol and only those nodes having sufficient energy (level 1 nodes in HET) requests information for making routing decisions. A node broadcasts a beacon-request packet to neighbors seeking location information. In response, a neighbor node sends back a beacon including its location via either a broadcast packet or a unicast packet as specified in the request packet. Bidirectional or unidirectional neighbors are discovered by the energy that remains in the nodes. Level 2 nodes have energy less than a threshold and hence they are assigned with unidirectional links. Level 1 nodes have energy more than the threshold and hence they are assigned with bidirectional links.

A forwarding node is always trying to select the neighbor closest to the destination as the next hop. The neighbor information is obtained from the HET. Forwarding fails when reaching a node, which has no neighbors closer to the destination. Then, the packet is forwarded using the tree constructed based on neighbor information of bidirectional links.

Data consistency problem: With the destination marked with location, a packet reaches the home node of the destination. Home nodes are categorized into two types: transient home node and persistent home node. For transient home node, it does not matter if the same data can be retrieved from the same location in different time. In contrast, data consistency is required for persistent home node, meaning that the same data sent to a location before should be retrieved later from the home node of the same location regardless the possible changes of the home node for the location.

A transient home node is identified when a packet reaches a node whose distance from the target location is less than half of its radio range and no neighbor nodes are closer to the destination. After the packet returns to the nearest node to the destined location and finds itself traversing a loop, the node is recognized as the home node for the location. For persistent home node, the first time a packet is sent to a location, HETRA identifies the target home node using the above method. Packets reach destination as they arrive at the marked home node for the target location without the traversal of the perimeter. Due to the dynamic nature of sensor networks, HETRA has special mechanism for persistent home node to keep the data consistency. The first time a node is identified and marked as a persistent home node for a location, it recruits all neighbors as replica nodes. Each replica node has a timer associated with it. The primary home node broadcasts refresh packets periodically to refresh timers on all neighbors.

When, the home node has energy less than the threshold, the home node is dead. Then, the timer in a replica node will expire and the replica node will keep sending a special packet to the target location reporting the death of the primary home node until receiving response. To handle the problem of new emerging home node for persistent home node, the current primary home node sends packets to the target location periodically to check the existence of new home node.

Implementation of HETRA and TCP/EXP in ns 2: HETRA is implemented as a new mobile ad hoc routing protocol in ns2. Since, ns2 is an open source, any additions can be incorporated to ns2 easily. The procedure to add new routing protocols is detailed in ns manual (Gavin and Vaidya, 2002). Three new programs that incorporated the structure of hetra are added to the ns2. These are hetra.cc, hetra.h and hetra.tcl. The program hetra.h includes basic header files needed for constructing a node with all basic attributes, to make the node a mobile node with the necessary attributes. These header files are already defined in ns2.

The program hetra.cc adds additional features such as constructing the hierarchical tree using the remaining energy of each node. The routing used is fixed routing but the hierarchical structure is constructed every 2 sec in the simulation performed in this study. But in practical applications the refreshing time for changing the hierarchical tree construction may be in the order minutes. This is due to the nature of application as the changes in sensed data may vary slowly and hence the change in energy consumed is also low.

The Program hetra.tcl provides the necessary mobility features needed for the mobile ad hoc nodes. This program also decides the characteristics of transmitter and receiver. This in addition to the routing provided by the hetra.cc defines the energy consumption characteristics of the HETRA nodes.

After including these files the ns2 is reinstalled so that these files get compiled and included with the basic routing algorithms already defined in ns2. The routing algorithm may be called by the tcl program very similar to DSDV, AODV and DSR. HETRA is the name to be used to call the new routing algorithm.

Similarly, the Exponential Congestion algorithm is implemented in ns2 by properly modifying the congestion control program tcp.cc. In tcp.cc all the previously available congestion control algorithm which use the basic AIMD rules, are implemented. Under these by setting the value of the variable windowOption_ as 11, the TCP/Exp algorithm with the following increase and decrease rules are added.

$$\begin{aligned} &\text{I: } W_{t+R} \leftarrow W_t + (\alpha/e^{W_t})^k \\ &\text{D: } W_{t+\delta t} \leftarrow W_t - (\beta\, e^{W_t})^l \end{aligned} \qquad (4)$$

By choosing different values of α and β in Eq. 4, they became the members of the exponential family. By including the higher order terms, exponential increase and decrease algorithms of different orders are obtained that may be used for the window size adjustment for the congestion avoidance phase. To use this option, the ns2 is reinstalled again. The tcl program by choosing the congestion control as TCP and windowOption_ as 11 will now make use of the exponential congestion algorithm.

RESULTS

For a sensor network to be useful, the location of each node must be determined (Howard *et al*., 2001). Fortunately, it is possible to determine the location of nodes in a network by conducting suitable simulation experiments and evaluating their efficiency. Simulation experiments were conducted aiming at evaluating the network lifetime and energy efficiency of the deployment patterns with a suitable routing algorithm and congestion control algorithm. Placement of sensor nodes plays a very important role on sensor networks (Howard *et al*., 2001; Bansal *et al*., 2002). Based on the position of the sensor nodes the efficiency of detecting the signal is improved (Clare *et al*., 1999). Hence, this study analyses five different distribution patterns of placing the nodes.

Node distribution patterns: Figure 2-6 shows, the five distribution patterns simulated Pattern 1 shown in Fig. 2, contain nodes placed in the form of a matrix with an area covering 500×250 m. Fifty nodes are placed at every 50 m

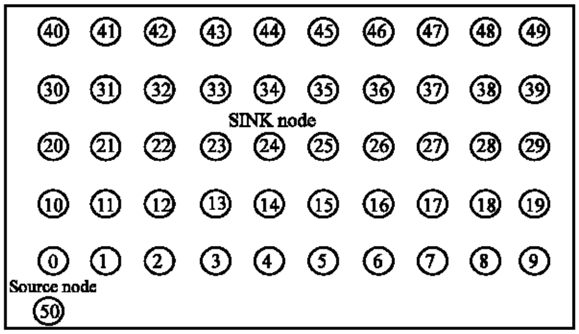

Fig. 2: Nodes placed at every 50 m in 500×250 m area (Pattern 1-ALL)

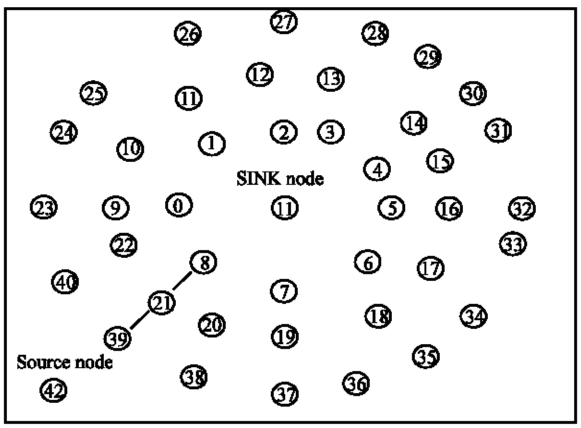

Fig. 3: Nodes placed as a step pattern in 500×500 m area (Pattern 2-CIRCULAR)

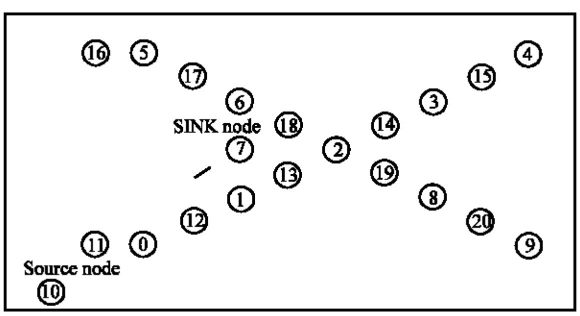

Fig. 4: Nodes placed in a cross pattern 500×250 m area (Pattern 3-CROSS)

distance in both X and Y directions. In this pattern at every cross section of the matrix a sensor node is placed. This type of pattern will normally have redundant nodes. But they are very efficient in detecting any mobile node within the entire area. A mobile source node is initially placed at (2, 2) and made to move through the area of

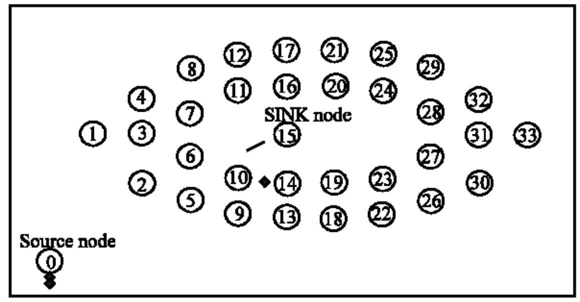

Fig. 5: Nodes placed in a cross pattern 500×250 m area (Pattern 4-EYE)

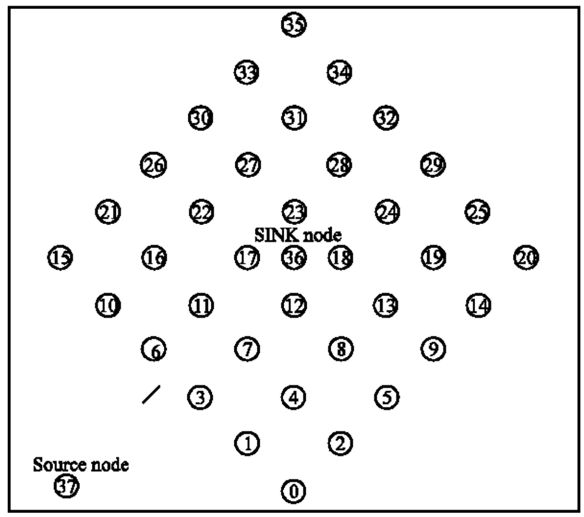

Fig. 6: Nodes placed in a cross pattern 500×500 m area (Pattern 5-DIAMOND)

500×250 m. A base station node, which is used to collect the sensing information is placed in a fixed position at (250, 150).

The Pattern 2 shown in Fig. 3 is a circular pattern. In this pattern the nodes are placed in positions such as to form the shape of the concentric circles. Totally, 42 nodes are placed for the entire area of 500×500 m. The source node is initially placed at (2, 2) and made to move through the area of 500×500 m. The base station node, which is used to collect the sensing information is placed in a fixed position at the centre of the circle i.e., at (250, 250).

The Pattern 3 shown in Fig. 4 is a placement of nodes as a 'X' (Cross). Twenty nodes are used for the area of 500×250 m and the source node and the base station nodes are place at the same position as the pattern 1.

Pattern 4, an eye pattern of placement of sensor nodes is shown in Fig. 5. The total number of sensor nodes in this pattern is 33. This pattern is simulated with an aim of covering more area than Pattern 3. Finally, Pattern 5 is created such that the nodes are placed in the form of a diamond. Totally, 37 nodes are placed to cover the area of 500×500 m. The source node is placed at an initial position of (2, 2) and to move within the area. The base station node is place at the centre i.e. (250, 250). This is shown in Fig. 6.

These five sensor nodes placement are used to evaluate their performance.

Simulation setup: The five patterns shown in Fig. 2-6 are simulated using ns2. Initially, the distance covered by a sensor node is evaluated by considering only two wireless nodes. With these nodes simulated as wireless nodes and by varying the antenna height and transmitter power, the minimum required antenna height and transmitter power required to cover a diagonal distance is found for each pattern. This power differs slightly for each node.

The various simulation parameters assumed are given below:

Channel	=	Wireless Channel
Propagation	=	Free Space
Network Interface Type	=	Wireless Physical Interface
Mac Type	=	Mac 802.11
Interface Queue Type	=	Drop Tail/Priority Queue
Antenna	=	Omni Antenna
Interface Queue Length	=	50
Routing Protocol	=	AODV/DSDV/DSR/HETRA
Antenna Height	=	0.35 m

With these parameters, the wireless sensor network is simulated with the five distribution patterns. In all the five scenarios, the source node placed initially at (2, 2), is made mobile. It is initially made to move towards the point (250, 200) and then towards (2, 500). The base station node receives the information from the sensor nodes.

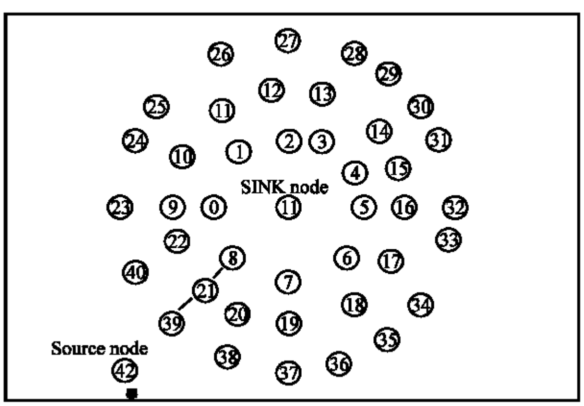

Fig. 7: Scenario generated by the simulation of Circular pattern generated by the nam

To analyze the energy efficiency and NLT of the wireless sensor network, the different routing protocols are assumed. They are AODV, DSDV, DSR and HETRA. Along with these routing protocols, five different TCP variants are assumed in the source node. They are TCP/Tahoe, TCP/Reno, MIMD-Poly, PIPD-Poly and TCP/Exp.

With the parameters mentioned, simulation is performed by transmitting packets from source node to the base station node through the sensor nodes. The simulation scenario generated the patterns is stored in a nam file. The nam display for one of the patterns is shown in Fig. 7.

DISCUSSION

The simulation program may include the following statements to generate the trace file.

```
set tracefd (open hetra.tr w)
```

The program used by ns2 to generate the trace is properly modified that will include the details such as energy spent and the remaining energy available after each transmission and reception. These trace files are properly utilized to extract the energy remaining in the nodes after each transmission and reception. The following AWK program performs this extraction.

```
set awkCode {
BEGIN { print "" >> "hetra1"; }
{if ($7 == "tcp" && $1 == "s" || $1 == "r")
{time = $2;
 energy = $14*1;}
 print time, energy >> "hetra1";}}
exec awk $awkCode hetra.tr
```

The above program extracts the information from the trace file generated (for example hetra.tr). The information extracted is the time (in seconds) at which the source node or the intermediate node either transmits ($1 = s) or receives ($1 = r) and the remaining energy (in Joules) in the source node or the intermediate node. After extracting the necessary details the energy versus time plot is drawn using the gnuplot program available in linux.

The trace file of simulation experiments conducted with the five different distribution patterns (ALL, CIRCULAR, CROSS, DIAMOND and EYE) with the four routing algorithms (AODV, DSDV, DSR and HETRA) and five congestion control algorithms (TCP, Reno, MIMD, PIPD, Exp) are used to find the NLT.

The values of NLTs extracted from the trace file of various simulations are depicted in Fig. 8-12. These figures give the variation of energy from initial 0.5 Joules

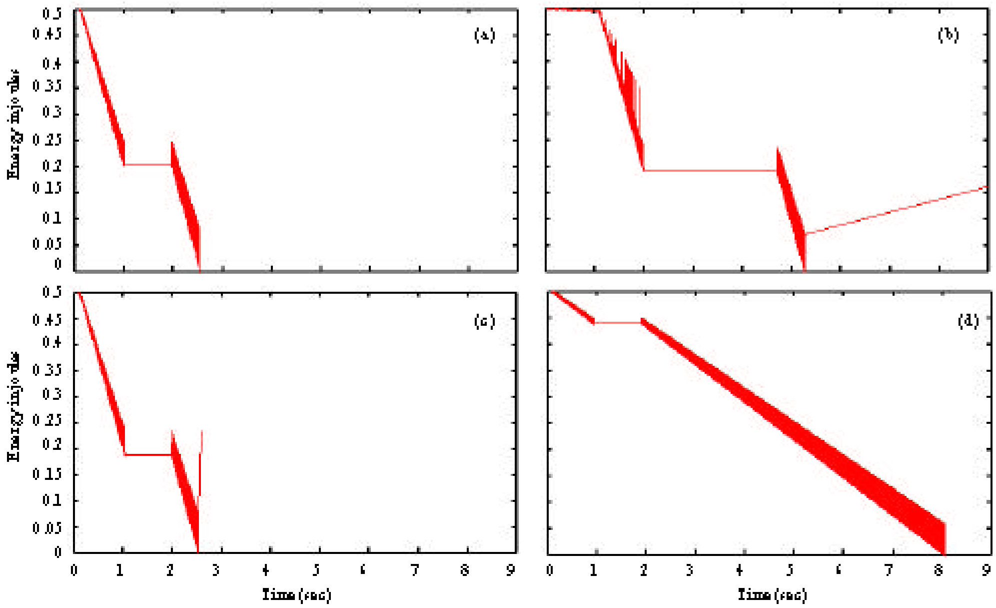

Fig. 8: Remaining energy in nodes of WSN with all pattern using various routing algorithms and TCP/Exp congestion control. Energy versus time plot of (a) DSR, (b) DSDV, (c) AODV and (d) HETRA

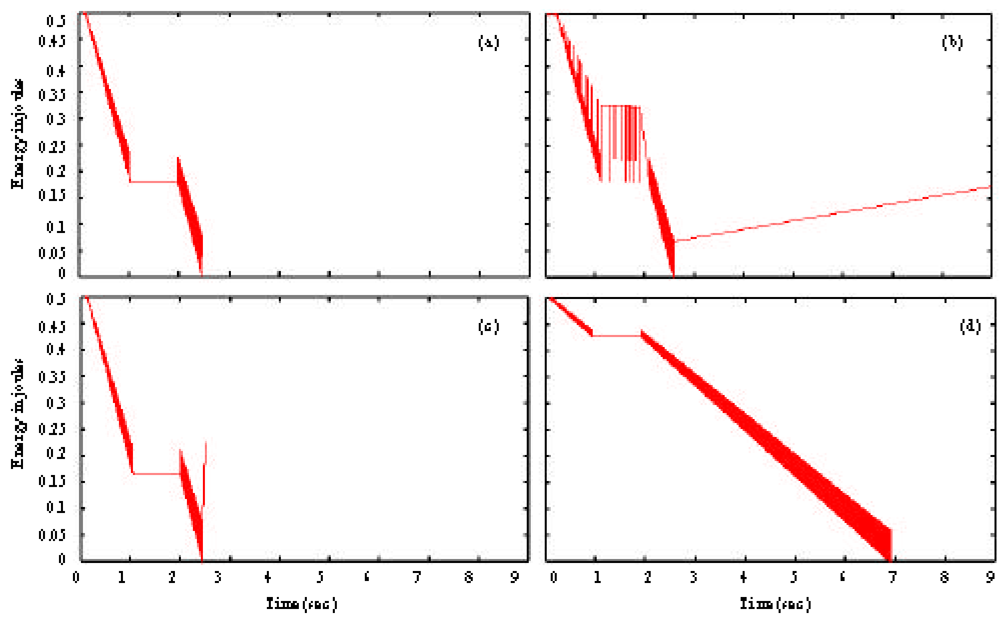

Fig. 9: Remaining energy in nodes of WSN with CIRCULAR pattern using various routing algorithms and TCP/Exp congestion control. Energy versus time plot of (a) DSR, (b) DSDV, (c) AODV and (d) HETRA

to zero Joules, in the duration of the NLT of each simulated network configuration. Assuming the TCP/Exp congestion control, Fig. 8 give the values of NLT for AODV, DSDV, DSR and HETRA routing protocols using the ALL node distribution pattern. Similarly, Fig. 9-12 give the NLTs using the other four node distribution patterns.

The NLTs of each simulation for all node distribution patterns is compared in Fig. 13. This figure shows that the hetra routing algorithm increases the NLT. This increase is due to the strategy used to select the nodes for transmission and reception based on the remaining node energy. The routing from source to destination is also

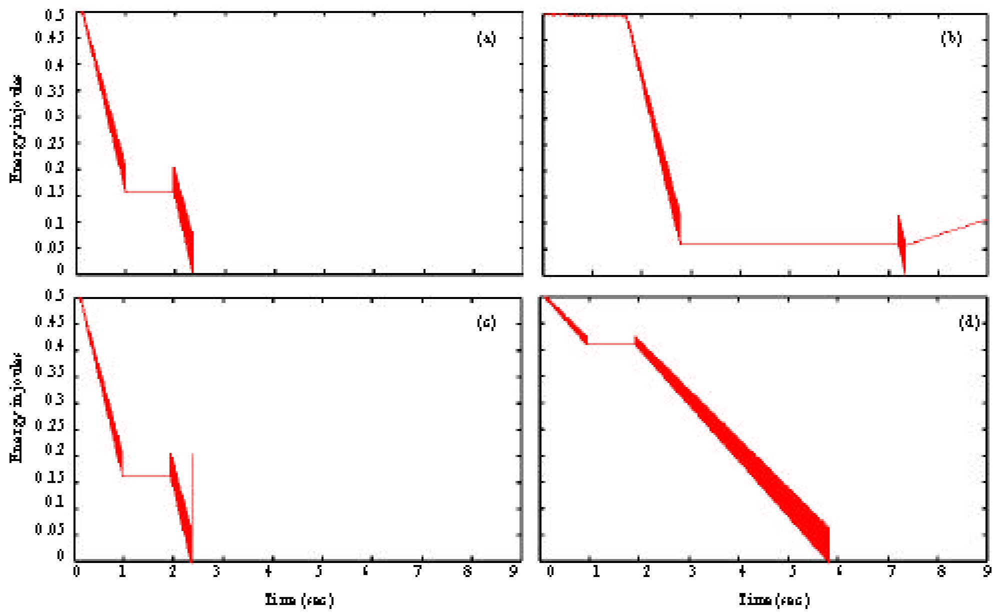

Fig. 10: Remaining energy in nodes of WSN with CROSS pattern using various routing algorithms and TCP/Exp congestion control. Energy versus time plot of (a) DSR, (b) DSDV, (c) AODV and (d) HETRA

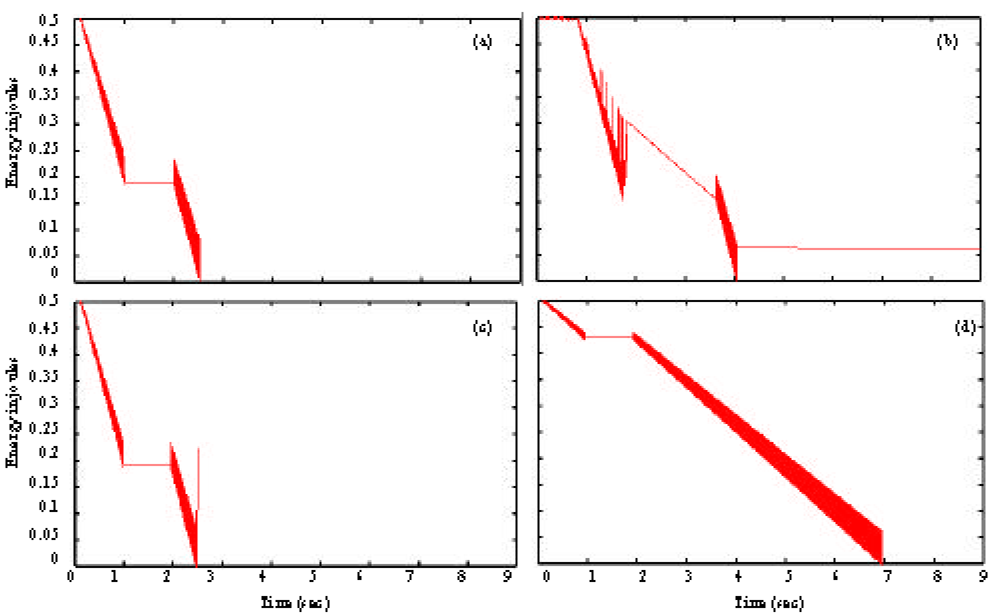

Fig. 11: Remaining energy in nodes of WSN with DIAMOND pattern using various routing algorithms and TCP/Exp congestion control. Energy versus time plot of (a) DSR, (b) DSDV, (c) AODV and (d) HETRA

based on the Hierarchical Energy Tree (HET) constructed using the above strategy. The exact values of NLT by assuming an initial energy of 0.5 Jules is depicted in Fig. 13. The numbers of packets delivered by these are also compared in Fig. 14.

This shows that the combination HETRA with TCP/Exp performs better. This combination has higher energy efficiency and also throughput. Hence, the NLT of WSN with HETRA/Exp combination has improved the overall efficiency of the network. As the number of nodes

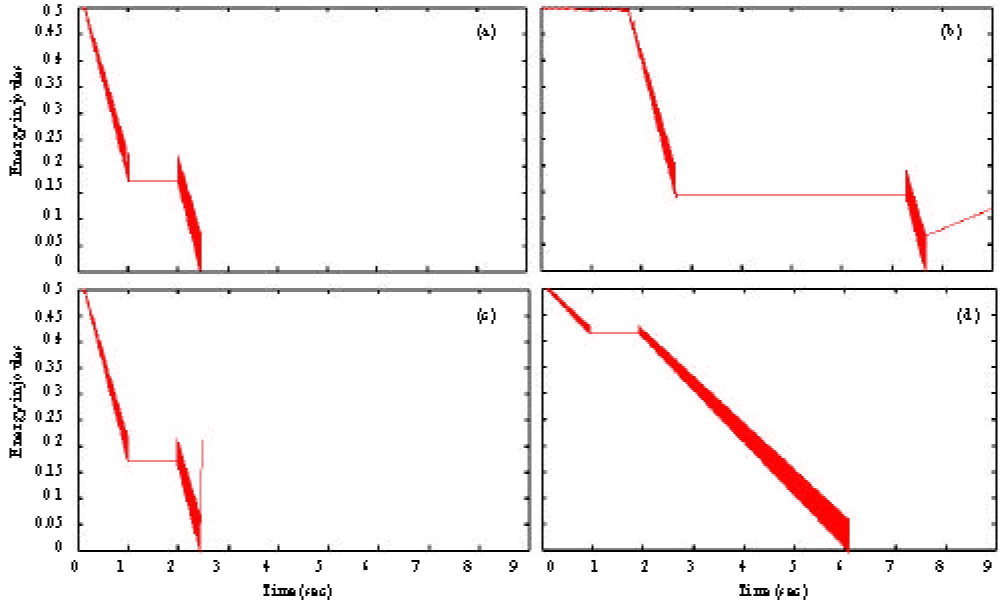

Fig. 12: Remaining energy in nodes of WSN with EYE pattern using various routing algorithms and TCP/Exp congestion control. Energy versus time plot of (a) DSR, (b) DSDV, (c) AODV and (d) HETRA

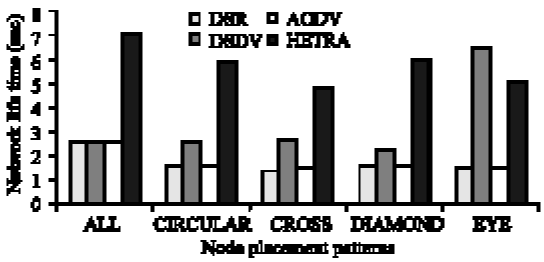

Fig. 13: NLT of WSN with various routing algorithms and TCP/Exp congestion control

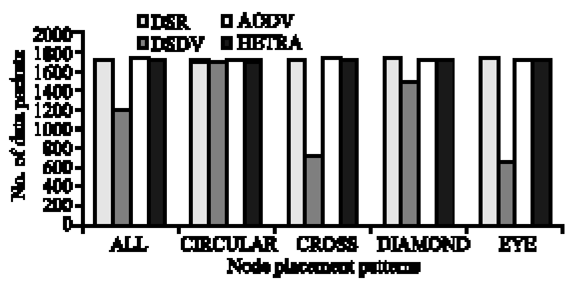

Fig. 14: Number data packets delivered in WSN with various routing algorithms and TCP/Exp congestion control

in CROSS pattern is less compared with other patterns and the energy efficiency is comparable with other patterns, the CROSS pattern is the most efficient pattern among the five fixed patterns assumed.

For comparing the performance of the different distribution patterns, this study aimed to formulate a common parameter. To arrive at a common parameter, initially Average Network Lifetime per node (ANLT) is calculated as given below.

Average network lifetime per node =
NLT/Number of nodes

This ANLT is calculated for all the simulated patterns. The result shows that the value is the maximum for CROSS pattern. Hence among the five patterns assumed, CROSS pattern with HETRA and TCP/Exp combination is the most efficient node distribution pattern of WSN.

In the CROSS pattern, the NLT with the congestion control algorithms Reno. MIMD and PIPD are higher than that of Exp. But, the number of data packets delivered is less than the Exp congestion control scheme. Hence TCP/Exp has higher throughput among the five congestion control algorithms.

Based on the number of data packets delivered and NLT a new parameter is formulated to compare the performance of WSN with the different node placement patterns, routing algorithms and congestion control algorithms. The parameter is named as Data packets-ANLT product and is evaluated as given below.

Data packets-ANLT (DANLT) product =
(Number of Data packets delivered) x (ANLT)

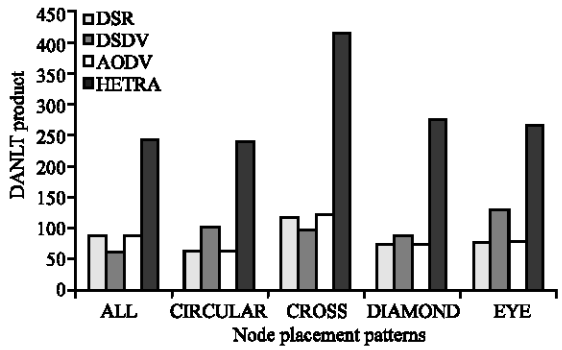

Fig. 15: Data packets-ANLT product of WSN with various routing algorithms and TCP/Exp congestion control

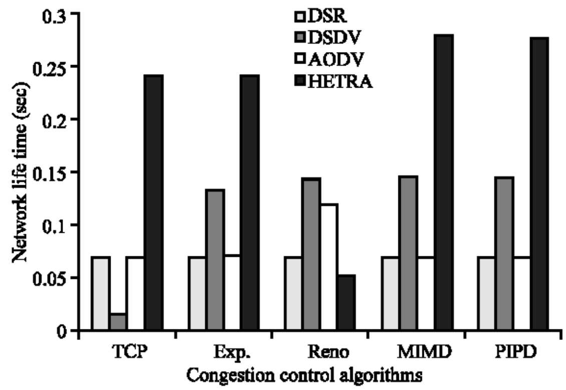

Fig. 16: NLT of WSN CROSS pattern with various routing and congestion control algorithms

This parameter gives an effective way of comparing the combined performance of energy efficiency, throughput, Network lifetime and number of nodes in each distribution pattern of WSN. This parameter is very useful in the comparison of the five node distribution patterns assumed.

The NLT of the five patterns (for TCP/Exp congestion control) are shown in Fig. 13. This shows that the NLT of WSN with HETRA and Exp combination is the highest among the four routing algorithms assumed. Among the patterns, ALL pattern has the highest NLT of 8.048 sec. The number of data packets delivered is compared for the five patterns (with TCP/Exp) in Fig. 14. This figure clearly indicates that the number of data packets delivered is the maximum for the combination of HETRA with TCP/Exp in cross pattern.

When Fig. 13 and 14 are independently analyzed, each comparison results in two different efficient patterns based on the NLT and data packets delivered. But if the parameter Data packets-ANLT (DANLT) product is evaluated, the result will clearly indicate the most efficient pattern among the five patterns. This comparison is depicted in Fig. 15. This figure clearly indicates that the DANLT product is the maximum for the combination HETRA with TCP/Exp in CROSS pattern. Hence, this combination proves to be the most efficient in terms of total energy spent, number of data packets delivered, Network lifetime with less number of nodes.

Similarly, in CROSS pattern the five different congestion control schemes are assumed and simulation is performed to compare the congestion control algorithm that best suites the WSN. Again the NLT for various congestion control schemes and routing algorithms are evaluated for the cross pattern from the trace file. The results are plotted in Fig. 16. Number of data packets delivered for the same combination of CROSS/HETRA with different congestion control schemes are also found from the simulation trace files and are shown in Fig. 17.

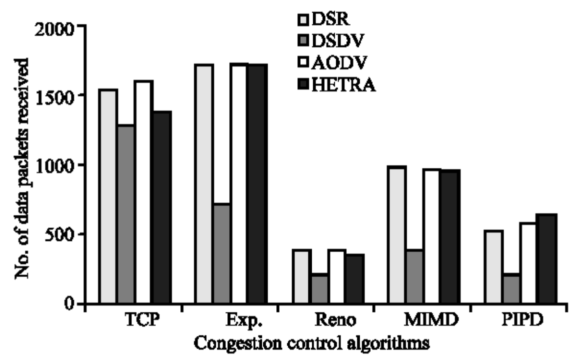

Fig. 17: Number data packets delivered in WSN CROSS pattern with various routing and congestion control algorithms

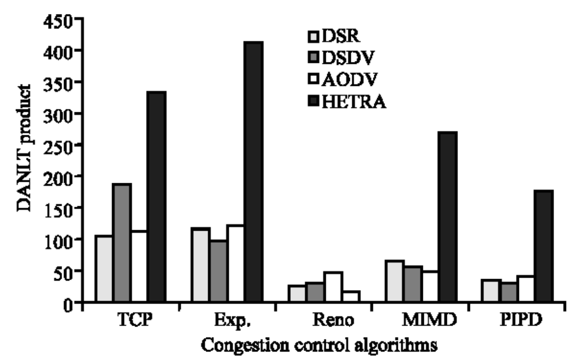

Fig. 18: Data packets-ANLT product of WSN CROSS pattern with various routing and congestion control algorithms

By looking at these figures, we could not infer a single combination of distribution pattern, routing algorithm and congestion control. This is because in Fig. 16, NLT is the maximum in HETRA with PIPD congestion control. The number of data packets delivered is the highest in HETRA with Exp congestion control as shown in Fig. 17.

Now the DANLT product is evaluated and shown in Fig. 18. This figure clearly indicates the combination CROSS pattern with HETRA and TCP/Exp is the most efficient among all the patterns.

CONCLUSION

This study aimed to analyze the NLT of WSN with different node distribution patterns assumed with the combination of routing algorithms and congestion control algorithms. The study assumed five patterns: ALL, CIRCULAR, CROSS, DIAMOND and EYE, four routing algorithms-DSDV, AODV, DSR and HETRA and five TCP congestion control schemes-TCP, Reno, MIMD, PIPD and Exp. Among these HETRA and Tcp/Exp are the newly proposed routing algorithm and congestion control algorithm.

Simulations are performed in ns2 with HETRA/Exp and TCP/Exp implemented in it. The results of simulation are used for the analysis. The trace files generated by the simulation experiments are used to evaluate the parameters such as NLT, ANLT, Data packets delivered and DANLT product. The NLT of each simulation with TCP/Exp congestion control algorithm, Four Routing Protocols and Five node distribution patterns are shown in Fig. 8-12. These parameters are compared in Fig. 13-18.

The results clearly indicates the WSN with the nodes distributed in the pattern of CROSS with the routing algorithm as HETRA and congestion control as TCP/Exp, performs most efficiently in terms of the energy consumption, network lifetime and throughput.

The future research will be on forming the efficient pattern by placing the nodes randomly and readjusting the positions by evaluating their DANLT product. Optimization techniques may also be used for maximizing the DANLT product by adjusting the node positions.

REFERENCES

Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. A survey on sensor networks. IEEE. Commun. Mag., 40 (8): 102-114.

Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler and John Anderson, 2002. Wireless sensor networks for habitat monitoring. In: ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta, GA.

Bansal, S., R. Gupta, R. Shorey, I. Ali, A. Razdan and A. Misra, 2002. Energy efficiency and throughput for TCP traffic in multi-hop wireless networks. In: Proc. IEEE. Infocom, pp: 210-219.

Chandrasekaran, M. and R.S.D.W. Banu, 2006a. Performance evaluation of polynomial congestion control algorithms MIMD-Poly and PIPD-Poly in TCP networks. Asian J. Inform. Technol., 5 (3): 346-357.

Chandrasekaran, M. and R.S.D.W. Banu, 2006b. Interaction between polynomial congestion control algorithms queue management schemes in wired TCP networks. Int. J. Soft Comput., 1 (2): 83-90.

Chandrasekaran, M., M. Kalpana and R.S.D.W. Banu, 2006a. Congestion Control using Polynomial Window size adjustment Algorithms for wired and wireless networks. In the Proceedings of International Conference on Network-ICN conducted at Mauritius, April. This study is now available in the Conference Proceedings by IEEE Computer Society and on IEEE Xplore. ISBN: 0-7695-2552-0. DOI: 10.1109/ICNICO-NSMCL.2006.67.

Chandrasekaran, M., M. Kalpana and R.S.D.W. Banu, 2006b. Interaction between MIMD-Poly and PIPD-Poly Algorithms and other TCP Variants in Multiple Bottleneck TCP Networks. In the Proceedings of IEEE Conference WOCN, Bangalore, April 2006. This paper is available on IEEE Xplore. ISBN: 1-4244-0340-5. DOI: 10.1109/WOCN.2006.166 6684.

Cheng, P., C.N. Chuah and X. Liu, 2008. Energy-aware node placement in wireless sensor networks. Int. J. Comput. Telecommun. Networking, 52 (11): 2189-2204.

Clare, L.P., G.J. Pottie and J.R. Agre, 1999. Self-organizing distributed sensor networks. In: Proc. SPIE Conf. on Unattended Ground Sensor Technologies and Applications, pp: 229-237.

Dhillon, S.S. and K. Chakrabarty, 2003. Sensor placement for effective coverage and surveillance in distributed sensor networks. In: IEEE. Wireless Communications and Networking Conference, Vol. 3, IEEE, pp: 1609-1614. ISBN: 0-7803-7700-1. DOI: 10. 1109/WCNC.2003.1200627.

Dunkels, A., T. Voigt, J. Alonso, H. Ritter and J. Schiller, 2004. Connecting Wireless Sensornets with TCP/IP Networks. In WWIC.

Gavin, H. and N.H. Vaidya, 2002. Analysis of TCP performance over mobile ad-hoc networks. In Wireless Networks, 8 (2/3): 275-288. DOI: 10.1023/A: 1013798127590.

Hou, Y.T., Y. Shi and H.D. Sherali, 2003. On energy provisioning and relaying node placement for wireless sensor networks. Technical report, Virginia Technology.

Howard, A., M.J. Matari´c and G.S. Sukhatme, 2001. Self localization in a distributed sensor network. Technical Report IRIS-01-407, Institute for Robotics and Intelligent Systems Technical Report, University of Sourthern California.

Kalpana, M. and T. Purusothaman, 2009. Performance evaluation of exponential TCP/IP congestion control algorithm. Int. J. Computer Sci. Network Security, 9 (3): 312-319.

Pan, J., Y.T. Hou, L. Cai, Y. Shi and S.X. Shen, 2003. Topology control for wireless video surveillance networks. In Proceedings of ACM Mobicom, ACM.

Wan, C.Y., A.T. Campbell and L. Krishnamurthy, 2002. PSFQ: A Reliable Transport Protocol For Wireless Sensor Networks. In WSNA, ACM 1-58113-589-0/02/0009.

Winfield, A.F., 2000. Distributed Sensing and Data Collection via Broken Ad Hoc Wireless Connected Networks of Mobile Robots. Distributed Autonomous Robotics Systems, Vol. 4. In: Parker, L.E., G.W. Bekey and J. Barhen (Eds.), pp: 273-282.

Wu, S.L., Y.C. Tseng and J.P. Sheu, 2000. Intelligent medium access control for mobile ad hoc networks with busy tones and power control. IEEE. J. Selected Areas Commun., 18 (9): 1647-1657.