

# High Survivable Routing Protocol in Self Organizing Wireless Sensor Network

Mohammad S. Al-Fares<sup>1</sup>, *student Member, IEEE*, Zhili Sun<sup>2</sup>, *Member, IEEE*, Haitham Cruickshank<sup>3</sup>, *Member, IEEE*

**Abstract**— Wireless Sensor Network (WSN) is one of the major research areas in computer network field today. The function of WSN in this paper is to provide sensing services in an un-attended harsh environment. Sensed data need to be delivered to the base station and to cope with the network unreliability problem. Few routing protocol takes into consideration of this problem. It is a great challenge of the hierarchical routing protocol to provide network survivability through redundancy features. In this paper, a short literature review of the existing routing protocol is carried out. Then a novel hierarchical routing protocol, which addresses network survivability and redundancy issues, is introduced. Initial analysis shows promising results of the proposed protocol comparing with LEACH, which is a well known protocol as benchmark. Finally, conclusion was drawn based on the research and future direction for further research is identified.

**Index Terms**— Wireless Sensor Network, Hierarchical, routing protocol, reliability, redundancy, survivability.

## I. INTRODUCTION

Wireless Sensor Network (WSN) is one of the major research areas in computer network field today. It is considered as one of ten emerging technologies that will bring far-reaching impacts on the future of humanity lives [1]. Also, the importance is due to that numerous applications can benefit from the WSN, such as healthcare, environmental, forest fire, and military applications, etc. [2-5]. As a new challenging research field, WSN now is undergoing an intensive research to overcome its complexity and constraints [5]. Such constraints are:

- Power source
- Communication and Bandwidth
- Mobility
- Processors and Memory
- Network density and Data aggregation

Because of these concerns, existing routing protocols cannot be deployed directly in WSN. Moreover, one of the

difficult challenging features to be offered by the routing protocol in WSN is to provide reliable network connectivity in the presence of harsh environment (resist to link and sensor nodes failures). The proposed routing algorithm presented in this paper is aimed to the forest fire monitoring and similar applications. The major specifications of these kinds of applications are the large area of deployment (about 20km x 20km), heterogeneous in sensed data and in Sensor Nodes (SN) types, and harsh environment (goes behind the high probability of node failure). Previous studies [6-8] have shown that link connectivity greatly affects the performance of routing protocols in WSN. However, the impact of unreliable link/node on the connectivity of WSN is not fully tackled in preceding research.

This paper presents a High Survivable routing protocol in Self Organizing WSN (HighSSO). It is designed to cope with the large area of deployment, link or node failures and heterogeneous network in forest fire monitoring and similar applications. The paper is structured by first a description of forest fire application in section 2. Then, section 3 briefly surveys the related works. Section 4 introduces the proposed hierarchical routing protocol. Section 5 presents the simulation tool. Results and performance analysis of the novel high survivable hierarchical routing protocol is presented in section 6. Finally, a conclusion and future directions in section 7.

## II. APPLICATION DESCRIPTIONS

This section gives a brief description of the forest fire and similar scenarios (for further information refers to [22] and [23]). Forest fire size is classified to small, medium, and large. The large size could reach more than 1 million acres [20]. The speed of fire front line is about 3~8% of the wind speed (depends on the density and type of vegetation, and slope). The shape of fire's front line takes a pear (Fig. 1) or elliptical like shapes.

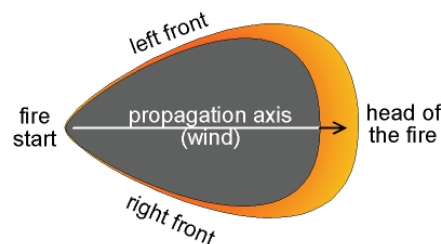


Fig. 1: Pear shape of fire's front line

1. Mohammad S. Al-Fares is a PhD Student in the Centre for Communication Systems Research (CCSR), University of Surrey, Guildford, GU2 7XH, Surrey, UK (phone: +44-1483-683-426; fax: +44-1483-686-011; e-mail: m.fares@surrey.ac.uk).

2. Zhili Sun is a full professor in the Centre for Communication Systems Research (CCSR), University of Surrey, Guildford, GU2 7XH, Surrey, UK (phone: +44-1483-689-493; fax: +44-1483-686-011; e-mail: z.sun@surrey.ac.uk).

3. Haitham S. Cruickshank is lecturer at the Centre for Communication Systems Research (CCSR) University of Surrey, Guildford, GU2 7XH, Surrey. Tel: +44 1483 686007, Fax: +44 1483 686011, e-mail: H.Cruickshank@surrey.ac.uk

Most of the large forest fires are contained in several weeks. The area temperature is high (reach more than  $92^{\circ}\text{C}$ ). Forest fire needs massive resources to fight it. It needs many fire crews, many helicopters, water tanks and pumps, trucks, civilians support and many more. All these resources can be organized and managed by the commander centre with the support of sensing data (see Fig. 2).

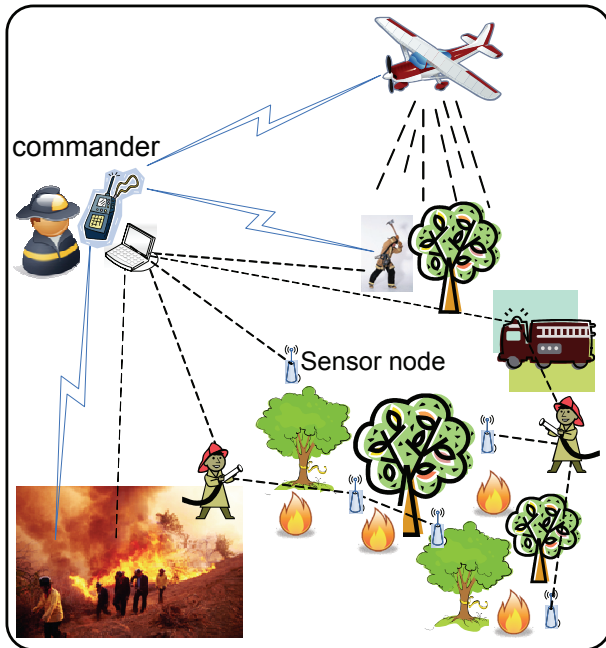


Fig. 2: Scenario of WSN in forest fire application

The huge damage caused by forest fire (reach multi-billion US\$), makes developing a new techniques or equipments a vital task. Based on the above, the WSN solution should work in a harsh environment with a heterogeneous data and devices. Fig. 3 shows one example forest fire in Kentucky State, US year 2007.

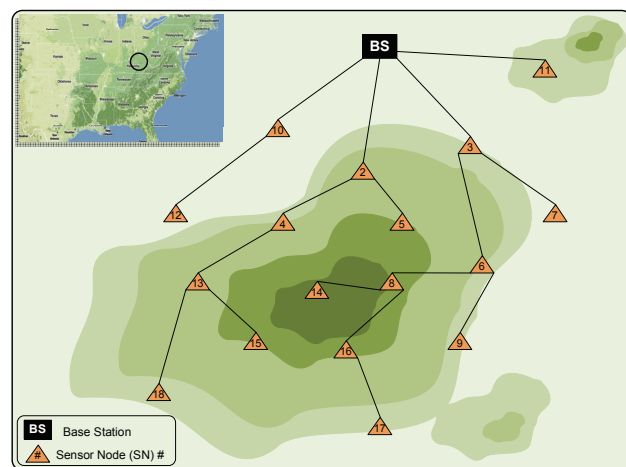


Fig. 3: Example of fire in US Kentucky State in year 2007

### III. RELATED WORK

Many routing protocols in WSN have been proposed to take into account of the inherent features of WSNs, along with the application and architecture requirements. These routing protocols can be classified according to the network structure, protocol operation, resource utilization, or routing protocols [2], [5]. Fig. 4 illustrates classification of routing protocols in WSN. The proposed routing protocol is classified under hierarchical network structure, and hybrid protocol operation that provides network connectivity and survivability features.

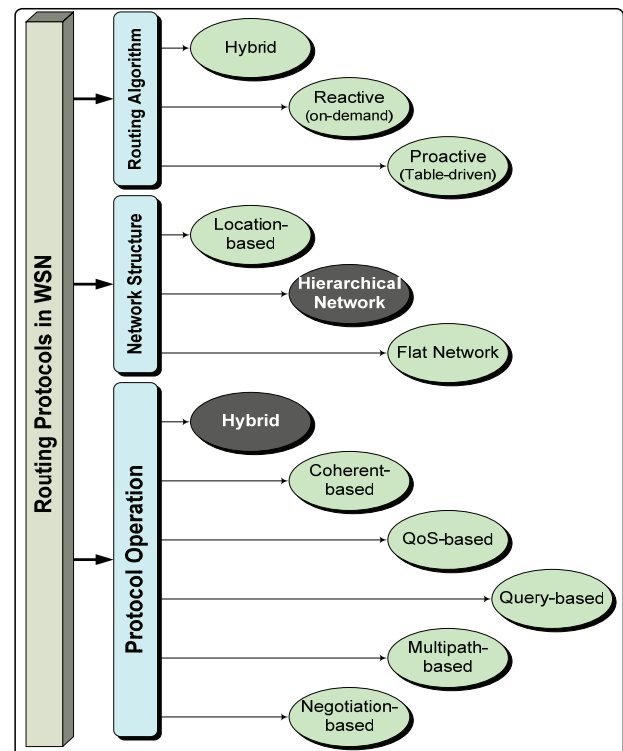


Fig. 4: Classification of routing protocols in WSN

Low Energy Adaptive Clustering Hierarchy (LEACH) Protocol [9] and LEACH-Centralize (LEACH-C) [10] are the well known routing protocols in WSN based on hierarchical structure. In general LEACH and LEACH-C divides the network into clusters (sections), and in each cluster there is an elected sensor node to act as head of the cluster identified as Cluster Head (CH). The cluster head task is to manage communication among member nodes of the cluster, data processing, and relay processed sensed data to the Base Station (BS) directly as seen in Fig. 5. LEACH and LEACH-C outperform flat network protocols in terms of network life time. However, LEACH and LEACH-C are not suitable to be deployed in a large area because of direct communication between CH and BS. Moreover, it does not take into account how the network resists the link and SN failure [5].

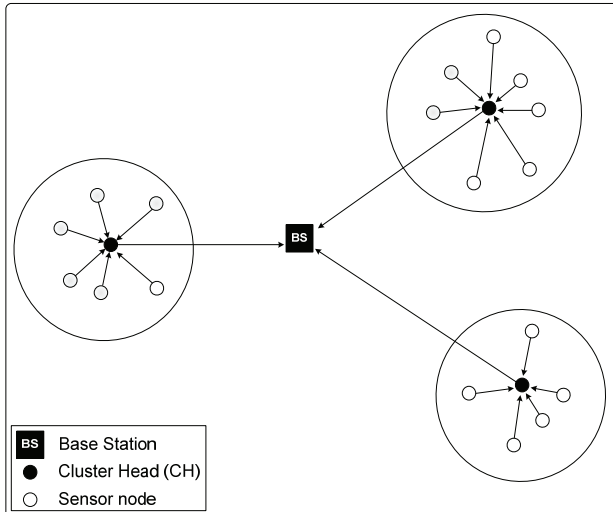


Fig. 5: LEACH network structure

Another hierarchical routing protocol is Self-Organizing Protocol (SOP)[11]. SOP routing protocol can be used with stationary or mobile sensor nodes. The protocol works simply by selecting a sensor nodes to act as router (relay messages), and these routers are to be stationary which form the backbone of the network to carry data to the BS. Every sensor node should be able to reach a router in order to be part of the network. A routing architecture that requires addressing of each sensor node is achieved, by identifying the address of the router node to which they are connected. A similar enhanced idea is the Proactive Routing with Coordination (PROC) [12], in which both protocols are used in a continuous dissemination (not on-demand) network type. PROC detect faulty node or link when a certain number of consecutive data is not acknowledged, then the sender node removes the failed node from its neighbor table and resend the data to the best selected parent node from the updated neighbor table. On the other hand, PROC have long recovery time (the time it takes to recover the failed node), and does not have control scheme for the traffic flow pattern in the entire network (local solution rather than global).

A different group of routing protocols is based on a spanning tree idea. The main protocols in this category are the Direct Spanning Tree (DST) routing protocol [13], Spanning Tree [14], Minimum Energy Spanning Tree (MEST) [15], Shortest Path Tree (SPT) [16], and Gathering Load Balanced Tree (GLBT) [17]. For this type of routing protocols, the network is formed by construction a tree covering the entire interest area rooted at the BS. The tree is constructed from selected nodes that meet specified criteria to be a member. Sender node will choose the best available nearby node to forward sensed data to it, and then the data will be directed in the same manner to the BS through the spanning tree. DST uses a technique of choosing the next parent node to route, based on the nearest node to the BS on a flat network. In general view, this technique is not necessary the best solution to forward data to the BS. Actually, it will cause the network traffic to be concentrated on certain nodes, especially nodes near the

BS, and prevents controlling the data distribution through the network (in case of congestion, it will cause higher delay and memory overflow). DST provides link or node failure by re-constructing the spanning tree.

A multi-hop hybrid routing protocol based on LEACH [18] is proposed by combining the clustering and multi-hop techniques (see Fig. 6). It uses a hierarchical multi-hop routing method to forward sensed data from CH to CH toward the BS. The protocol does not provide redundancy and recovery techniques in case of link or node failure, but for the next cycle of electing a new CH the fault will be resolved automatically (takes long time) for in-cluster communication. For intra-cluster communication if the intermediate CH or communication link failed then the CH will search for alternative path. In case of no alternative path exist, then the CH will send the data directly to the BS which is not feasible solution in large area of deployment.

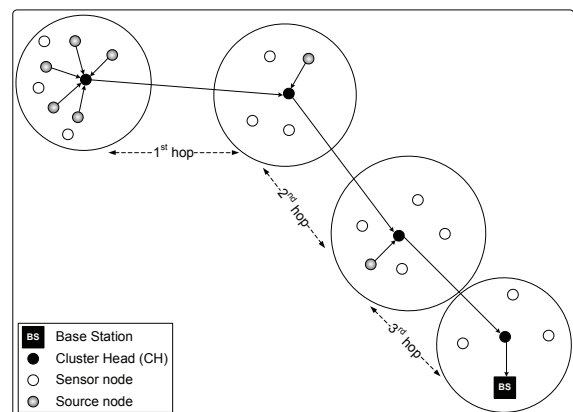


Fig. 6: Multi-hop hybrid routing protocol based on LEACH

#### IV. HIGH SURVIVABLE ROUTING PROTOCOL IN SELF ORGANIZING WIRELESS SENSOR NETWORK

In this section, we present a new routing protocol. One of the key points in the proposed routing protocol is to solve the link and node reliability problem. Most protocols described in section III especially hierarchical do not address link/node reliability problem. It is a challenge for the hierarchical routing protocol to provide network survivability and redundancy features. Therefore, HighSSO routing protocol designed to cope with these features.

In general, HighSSO uses multi-hop hierarchy (to cover large area) and spanning tree (for fast routing and less overhead) as basic ideas to deal with large area of deployment issue. HighSSO is fully distributed that every CH choose the nearest parent-CH to the BS to forward data to in the normal situation. If the network is congested or the parent-CH is dead, then the CH chooses the next best parent with zero communication (because of the wireless broadcast communication nature). Furthermore, HighSSO takes high density SN deployment advantage, to provide network fault tolerance feature through introducing redundant CH. Mainly the operation of

HighSSO can be divided into three parts. The start-up phase, the setup phase and the process phase. Fig. 7 illustrates the proposed algorithm.

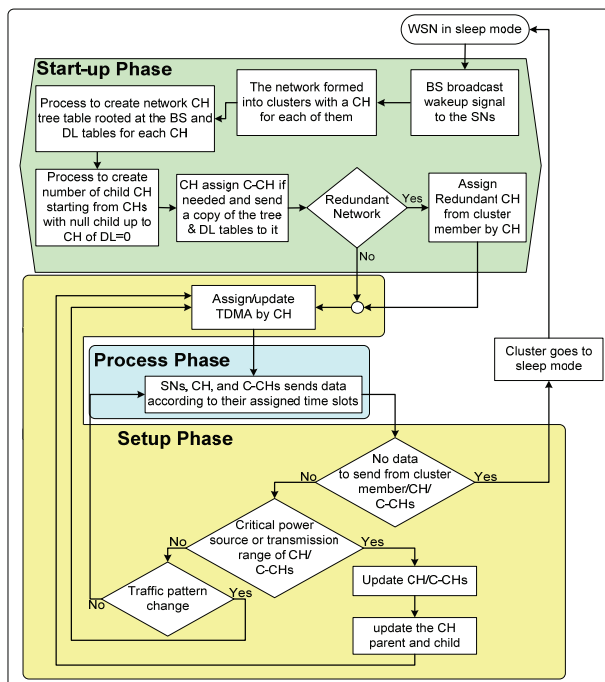


Fig. 7: Flow chart of the proposed algorithm

#### A. Start-up Phase

After deploying the SNs in the area of interest in a sleep mode (could be deployed long time ago), the start-up phase begins. This phase is responsible to build up the WSN in the network layer. It starts by firstly BS send a wakeup signal to the network. Then the network will be formed into clusters each with assigned elected CH (initially same as in LEACH [9], and then afterwards remaining power source is the added factor). Subsequently, a tree is created from the BS as the root up to the leaves through only the CHs (this process will be described on the next paragraph). An example of this process can be seen in Fig. 8.

Based on the tree construction, there will be parents and children for each CH except for the leaf CH where it will be parents only. For every turn of electing a CH of a cluster, the new CH will send a control message to its parent and child(s) (if exist) to update the changes of the tree members. After that, the total number of children of each CH will be specified starting from leaf CH to its parents CH up to the root (BS). Subsequently, we can benefit of the network high density deployment and the inexpensive cost of the SNs, by helping the CH to relay and process data of other children clusters toward the BS. This procedure is via SN of cluster to be as Co-operative cluster head (C-CH), and assigned by the CH according to the next rank on the CH election. This procedure is applied to CH with a total number of children exceeds  $n$  (from simulation, optimal  $n$  will be chosen). This will help the

network to avoid power drain from the CH. Especially for the CH near to the BS (energy holes problem [19]). In addition, the procedure provides more paths to the BS which helps in solving the data converge-cast problem. Converge-cast is a communication pattern, where the data flow is from a set of nodes to one node (many-to-one). This point will make the procedure simple (not complicated), that the code is small which helps in saving memory space and needs less processing power. Finally, redundancy option for the CH is specified by the BS to be either fully-redundant, or semi-redundant, or no-redundant. Fully-redundant is simply assigns a redundant CH node by means of the CH for each cluster. In semi-redundant, the CH near to the BS will assign a redundant node and the CH distant from the BS will not.

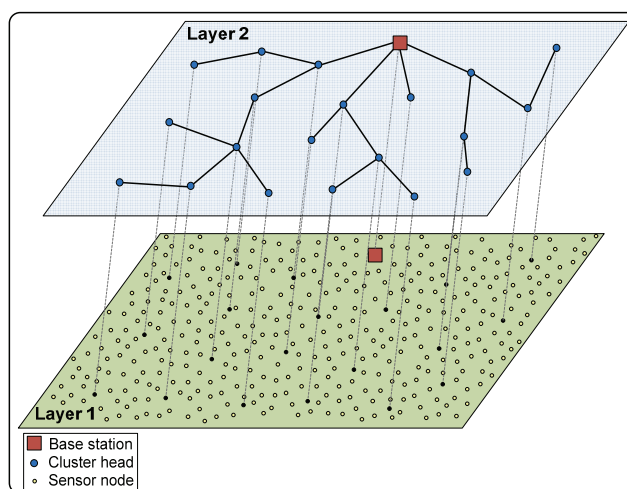


Fig. 8: Example of tree formation process

#### Tree Creation Process

The tree creation process initially starts from the BS. The process uses three-way messaging technique as shown in Fig. 9.

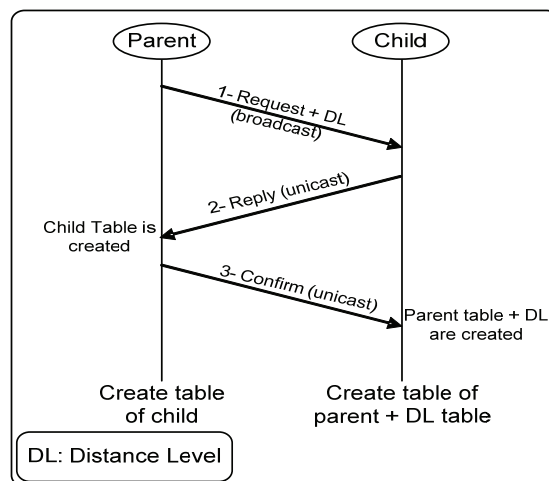


Fig. 9: Tree creation process using three-way messaging techniques

The tree will be created as follow:

- BS (afterward parent CH) send a tree formation (TF) message to its neighbour CH(s) that are in the modified transmission range.
- The neighbours CH(s) reply with tree join (TJ) message to the BS asking to add them to the tree.
- A confirmation tree join grant (TJG) message is sent from the BS to the replied CH(s) associated with its distance level value. Then BS create a table of all answered back children CH(s). As well, the replied neighbour CH(s) will create a parent table that contain the parent and the BS with distance level of zero. The Distance Level (DL) is the number of hops from CH to reach the BS via intermediate parent CHs, and is equal to the  $DL_{\text{parent}} + 1$ . Afterwards, CH may have many parents, and then forward data to the parent with minimum DL. This will help to transfer data to other route if one link is failed or congested, by maintaining the DLs in the parent table (increase DL of parent by 2 if it congested). Maintaining DLs of the parent table will achieve network reliability feature.

As soon as the CH join the tree, it will repeat the 3 ways messaging process (steps a to c). If the CH(s) receives no reply then it will stop the operation and knows that it is the leaf CH of the tree. Fig. 10 shows the flow chart of the tree formation packet.

#### ▪ Number of child process

This process helps each CH to discover the expected load by find out number of clusters to be served (i.e. number of child CH(s) down to the leaf tree).

The leaf CH will start the process by sending a number of child (NC) message to its parent. Where the NC field in the message is set to zero (has no child). The parent will then update its child table. After that, the parent CH checks if all child reply with a NC message. If so, then it sends a NC message to its parent with the total number of grandchild CH. The process is continuing bottom to top of the tree until reaching the BS.

#### B. Setup phase

In this phase, every CH will manage communication time schedule of its cluster SNs member. In every cycle, CH will assign communication time slot to every cluster members according to their needs. This allows SNs to identify when to wakeup to send its sensed data to the CH to save energy (extend network life time).

#### C. Process Phase

This phase has the longest time interval compared to the start-up and setup phases. It is the event where the sensor nodes send their requested sensed data to the BS via CH/C-CH according to their time slots. As well, CH and C-CH relay processed data (remove duplicate data,

enhance data accuracy, and compress data of similar priority) to the BS.

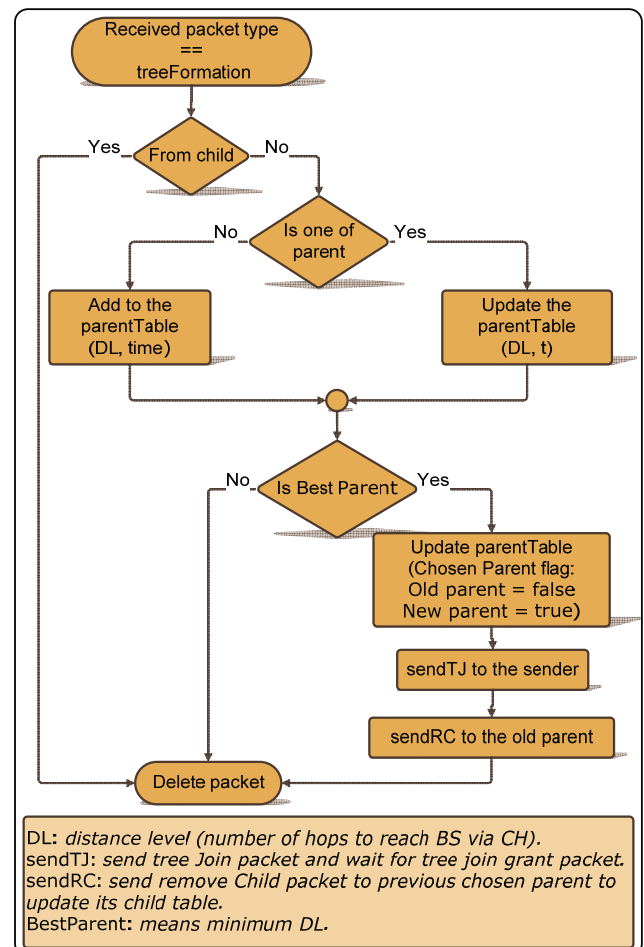


Fig. 10: Flowchart of the tree formation message

## V. SIMULATION TOOL

There are many simulation tools available as commercial or open source software. But none of the existing is ready to be used in our application. Therefore, we need to build and code the proposed routing protocol with the application scenario. After examine the available simulation tools we find that the OMNeT++ with INET framework is more suitable for our simulation studies.

OMNeT++ is an open-source discrete event simulation environment. Its primary application area is the simulation of communication networks. OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, then assembled into larger components and models using a high-level language. Reusability of models is one of its advantages. In addition, one of our decision reasons is that OMNeT++ has extensive GUI support, and due to its flexible modular architecture, the simulation kernel (and models) can be embedded into our application. The INET Framework is an open-source communication networks simulation package for the OMNeT++ simulation

environment. The INET Framework contains models for several Internet protocols: UDP, TCP, IP, IPv6, Ethernet, PPP, IEEE 802.11, OSPF, and several other protocols.

We install the OMNeT++ 3.2 with the INET framework on Microsoft (MS) Windows XP SP3 platform. MS Visual Studio 2005 (VC++) is installed with MS Platform SDK for Windows Server 2003 SP1. Furthermore, OMNeT++ requires a TCL/Tk script language with BLT extension. And finally, Ghostscript 8.62 was installed for the postscript graphs. Fig. 11 illustrates the coding structure under the OMNeT++ platform.

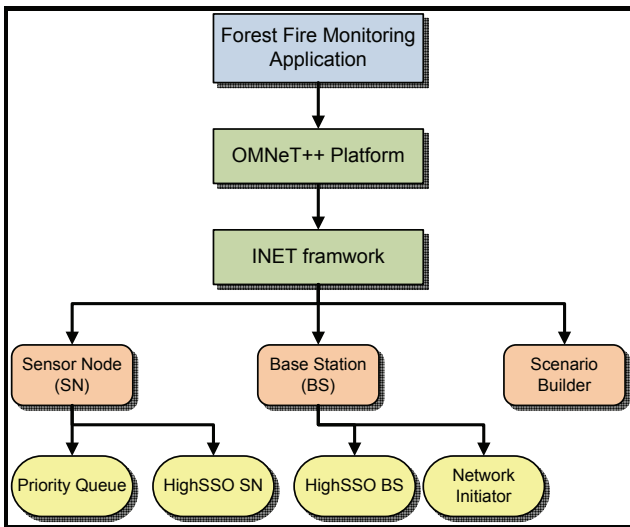


Fig. 11: Simulation code structure

## VI. RESULTS AND ANALYSIS

To study the proposed protocol in network connectivity and survivability features, we need to answer the question. How long does the network take to recover from faulty link or node? And how much power it consumes?

To be able to answer this question we use OMNeT++ 3.2 simulation tools with INET framework. We consider the same power model used in [9]. Where the transmission power ( $P_{tx}$ ) and reception power ( $P_r$ ) are

$$P_{tx}(k, d) = \begin{cases} k * E_{elec} + (k * e_{fs} * d^2), & d < d_o \text{ (short\_distance)} \\ k * E_{elec} + (k * e_{mp} * d^4), & d \geq d_o \text{ (large\_distance)} \end{cases}$$

$$P_r(k) = k * E_{elec}$$

Where:

$E_{elec}$  = energy consumption to run the transmitter/receiver.

$e_{fs}$ ,  $e_{mp}$  = radio energy dissipation, which consumed in the transmitter amplifier in free space (fs) and multipath (mp).

$k$  = number of bits to be transmitted.

$d$  = distance between transmitter and receiver in m.

With the same parameters of  $E_{elec} = 50$  nJ/bit and  $e_{amp} = e_{fs} = e_{mp} = 100$  pJ/bit/m<sup>2</sup>. We assume that CH will send 400 bits/packet. The simulated area is 1500m x 1000m, with a random number of SNs deployed randomly.

Also, the scenario is created using C++ to specify random failure by using Poisson's reliability SN distribution [21].

$$R_n(t) = e^{-\lambda_n t}$$

Where:

$\lambda_n$  = the failure rate of sensor node n (assumed 15%).

$t$  = the time period.

The assumptions for the network in our study are:

- Fixed location of BS, CHs, and redundant CH.
- The SNs are equipped with the power control facilities to vary their transmitter power.
- The clusters assumed to be circular and then the transmissions take a spherical shape.
- Assume worst case failure (node failure).

From the simulation, Fig. 12 shows a comparison in terms of total power consumption. It shows that both HighSSO types consume less energy than LEACH. Moreover, the difference in power consumption is increased as the number of sent packets increased. In Forest fire and similar applications, LEACH could not far communicate directly with the BS, but in simulation we assume it is possible (if new wireless technology found). Also, we use the short distance of  $P_{tx}$  as well, otherwise the power consumption in LEACH will be higher.

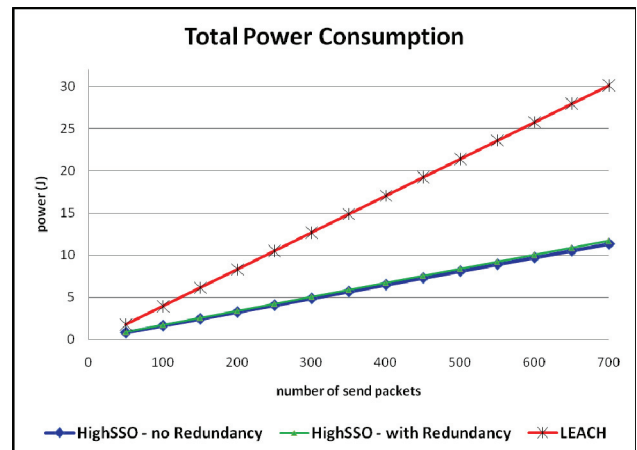


Fig. 12: Total power consumption

On the other hand, the response time for the HighSSO in case of failure can be clearly revealed in Fig. 13. It shows end-to-end delay per packet with 8 hops to reach the BS. Where, the CH failure occurs during sending packet 21. HighSSO with redundancy respond better and faster by about 1/3 less time than HighSSO with no redundancy.

One of the forest fire application requirements is to route heterogeneous sensed data (quality of service (QoS)). The data has been categorized using a priority field on the packet header. In which enable HighSSO to serve the packets with higher priority before the lower one. The classification comes from the application nature. This

feature makes HighSSO to operate on heterogeneous WSN network.

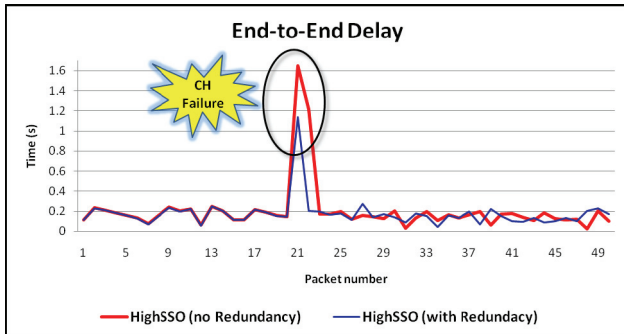


Fig. 13: HighSSO delivery time comparison

By using simulation we investigate the HighSSO QoS behavior with respect to the end to end time delay. From the simulation Fig. 14 shows the end-to-end time delay of heterogeneous data with a single path to the BS. The figure illustrates that the time sensitive data gets guaranteed service response even if the network is busy with normal data. Also, the figure is build up based on no alternative path to the BS is available. Regardless to the rise of time delay for the normal data, HighSSO guarantee data delivery to the BS. This helps in extending network life time by avoiding resend the same data.

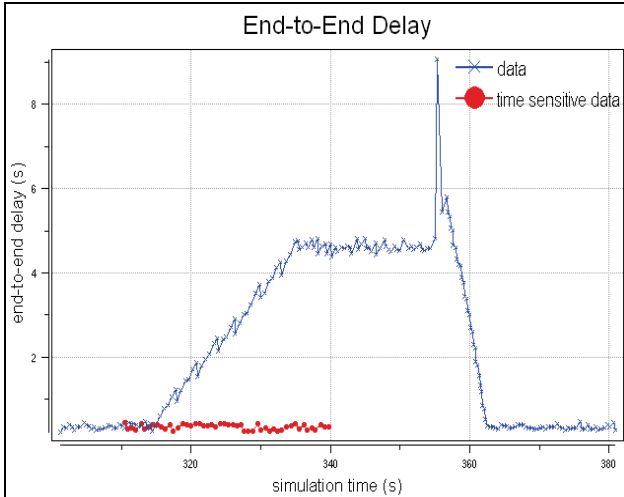


Fig. 14: End-to-end delay of heterogeneous data with a single path to the BS

In addition, if one path is used from source to BS in multi-hop technique, it will leads to extensive energy consumption to this path. With the time this results to an un-even network energy distribution.

HighSSO routing protocol take care of this issue by distribute the traffic on the network if possible. This can be seen in Fig. 15, if the normal data congested with a time sensitive data, then it try to search for alternative path. This

will help in improving the end-to-end delay of the normal data and to distribute the traffic throughout the network.

Finally, Fig. 16 illustrates the HighSSO performance on the whole network with multiple senders. It can be seen that HighSSO guaranteed the priority service to the time sensitive data than the normal one. Additionally, it avoid congestion path using alternative path if available.

Besides simulation results, a further Analysis is applied using the following example. The results of the analysis are used to generalize the solution. Let's assume the scenario sample shown in Fig. 17 in which CH7 try to send data to the BS.

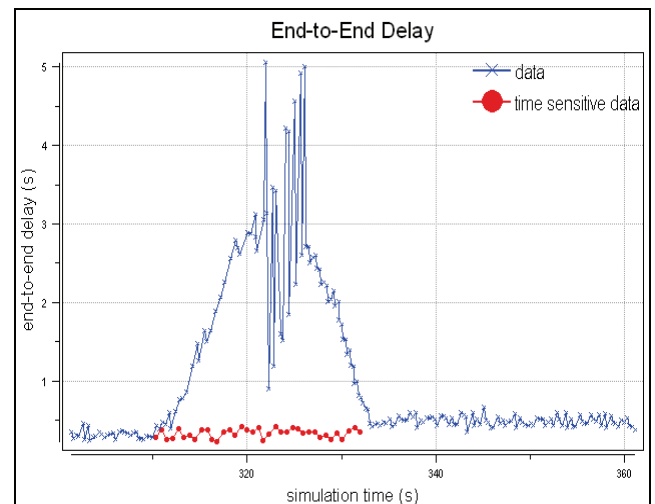


Fig. 15: End-to-end delay of heterogeneous data with several paths to the BS

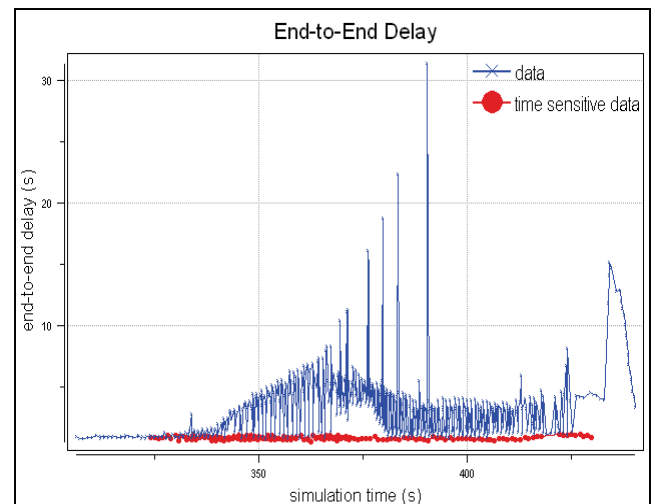


Fig. 16: End-to-end delay of heterogeneous data of the whole network

#### A. Recovery time in case of link failure ( $R_L$ )

The time to send a message between children and parent CHs is

$$T_m + T_r + T_{\text{delay}} \quad (1)$$

Where:

$T_m$ : is a time for a message to travel from sender to receiver.

$T_r$ : is a time to process the received message plus the time required to travel the reply message to the sender.

$T_{delay}$ : is a safety margin and is small compared to  $T_m$  and  $T_r$ .

As an example, if the sender (CH<sub>7</sub>) didn't receive a reply for the second time then it will update its parent table (see TABLE 1) and send the same message to the new parent (CH<sub>6</sub>) with the minimum of DL.

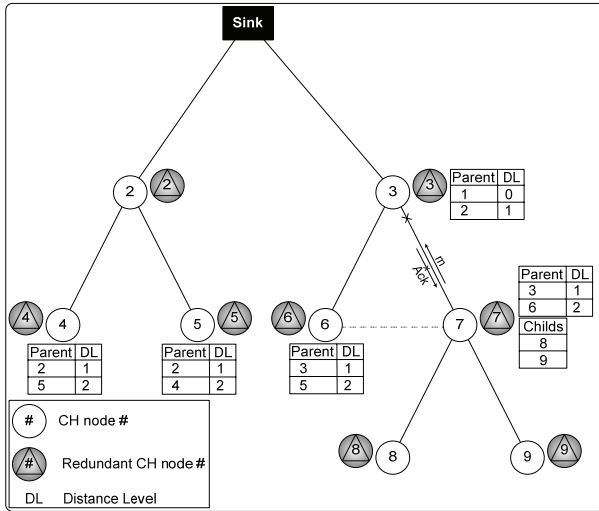


Fig. 17: Example of tree scenario

TABLE 1: CH7 PARENT TABLE

| Parent | DL       |
|--------|----------|
| 3      | $\infty$ |
| 6      | 2        |

The time to detect the broken link or CH failure is

$$2(T_m + T_r + T_{delay}) \quad (2)$$

CH<sub>7</sub> use a 2-way message to join the new parent as a child. This process takes time of

$$(T_{request} + T_{confirm} + T_{delay}) \quad (3)$$

Where:

$T_{request}$ : is the time for a request to travel from sender (CH<sub>7</sub>) to receiver (CH<sub>6</sub>).

$T_{confirm}$ : is the time to process the received message and a confirm travel time from receiver (CH<sub>6</sub>) to the sender (CH<sub>7</sub>).

The message sent to CH<sub>6</sub> takes time of  $(T_m + T_r + T_{delay})$ . Then CH<sub>6</sub> will send the message to CH<sub>3</sub> in a time of  $(T_m + T_r + T_{delay})$ . This will lead the time of choosing alternative path, which is

$$2(T_m + T_r + T_{delay}) + (T_{request} + T_{confirm} + T_{delay}) \quad (4)$$

Then the time to recover a failure link is

$$= \begin{cases} [4(T_m + T_r + T_{delay}) + (T_{request} + T_{confirm} + T_{delay})] ; \text{ same level of DL} \\ [3(T_m + T_r + T_{delay}) + (T_{request} + T_{confirm} + T_{delay})] ; \text{ parent level of DL} \end{cases} \quad (5)$$

Similarly we have the general formula as follows

$$R_L = [n(T_m + T_r + T_{delay}) + (T_{request} + T_{confirm} + T_{delay})] \quad (6)$$

$; n(\text{integer}) \geq 3$

If  $n=3$  it means recovered to parent level of DL to the BS.

If  $n=4$  it means recovered to same level of DL to the BS.

In LEACH if the link fails (disconnected), then the communication will breakdown because there is a direct transmission between CH and the BS.

#### B. Recovery time in case of node failure without redundancy ( $R_N$ )

The same analysis done in the link failure case is used to find the recovery time for the case of node failure. Discovery time of (parent) or link failure is equal to

$$2(T_m + T_r + T_{delay}) \quad (7)$$

And the discovery time of (parent) failure is equal to

$$5(T_m + T_r + T_{delay}) + (T_{request} + T_{confirm} + T_{delay}) ; \quad (8)$$

if it re-route to node that have same parent

Then the time to choose alternative path is greater than or equal to

$$[3(T_m + T_r + T_{delay}) + (T_{request} + T_{confirm} + T_{delay})] \quad (9)$$

Another discovery method is by repeating the election process. If the CH is dead then it will not involve into the next election process.

The time to replace the failed CH is

$$T_{election} - (\Delta t_{e-process} + t_{elec}) ; \quad (10)$$

$0 \leq t_{elec} \leq (T_{election} - \Delta t_{e-process})$

Where:

$T_{election}$ : schedule election cycle time.

$\Delta t_{e-process}$ : election process time.

$t_{elec}$ : live time after election process.

Then in general

$$R_N = \text{Minimum} \begin{cases} 3(T_m + T_r + T_{delay}) + (T_{request} + T_{confirm} + T_{delay}) \\ T_{election} - (\Delta t_{e-process} + t_{elec}) \end{cases} \quad (11)$$

The effect of node failure is more than link failure. That affects its entire children and parent attached to it and need to be replaced with a new node. Where as, the link failure is less frequent to happen and only affect one node attached to it by the failed link.

In LEACH, the node failure will not be discovered and replaced only on the next CH election process. Then,  $R_N$  is equal to equation (10).

It can be seen our new protocol perform better than LEACH in this case.

### C. Recovery time in case of node failure with redundancy ( $R_{N,r}$ )

In critical application, redundancy is required in the case of node failure. Actually, Redundancy takes place on the expense of the network SN resources. This action is acceptable due to the high density deployment and the low cost of SNs. Also, it can be compromised by using semi-redundant network, in which the CH near to the BS is more critical which needs redundancy. In this analysis, we will assume full-redundant. Redundant CH (R-CH) node works exactly like CH, but without respond (silent respond). So, if R-CH listens to a message sent to its CH without respond then it will respond immediately to the second time the message is sent and then take over the CH operation and choose a new R-CH from its cluster members. This process takes  $2(T_m + T_r + T_{\text{delay}})$ . Or if the R-CH detects silent operation during CH slot in the cluster TDMA then it will take over the failed CH. The failure detection time is

$$T - t_i; \quad 0 \leq t_i \leq (T - \Delta t) \quad (12)$$

Where:

$T$ : is the complete cycle time.

$\Delta t$ : is the time slot.

$t_i$ : is the life time of CH from the next TDMA of its time slot to the last time slot before the CH slot time.

$$\text{The time to takeover the failed CH is} \\ (2T - \Delta t - t_i) \quad (13)$$

$$\text{Generally, the time to recover a failure CH node is} \\ R_{N,r} = \text{Minimum} \begin{cases} 2(T_m + T_r + T_{\text{delay}}) \\ (2T - \Delta t - t_i) \end{cases} \quad (14)$$

Whereas LEACH does not provide redundancy in such situation.

For comparison assume  $T_m = T_{\text{request}} = 130$  ms,  $T_r = T_{\text{confirm}} = 170$  ms,  $T_{\text{delay}} = 20$  ms, and  $T_{\text{election}} = T = 10$  s (parameters taken from simulation). Then the response time for HighSSO with no redundancy  $R_N = 1280$  ms. Whereas, the response time for HighSSO with redundancy  $R_{N,r} = 640$  ms. Fig. 18 shows the response time of the two mode of HighSSO which match the above calculations.

In conclusion, we can see the proof of analysis through the simulation, that the relation between  $R_N$  and  $R_{N,r}$  is reflected correctly on Fig. 13.

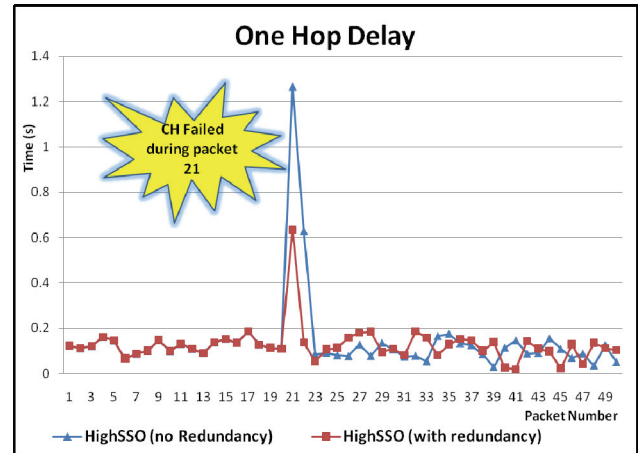


Fig. 18: Network Response Time with failure at node 7

## VII. CONCLUSION AND FUTURE DIRECTIONS

This paper has shown the importance of network survivability and connectivity to overcome the link and node failure. The proposed protocol provides features of network survivability and redundancy. Initial analysis shows that HighSSO overcomes the weakness of the LEACH in recovery of link and node failures. In addition, simulation shows that HighSSO consumes less energy than LEACH. And we demonstrate that HighSSO is capable to deal with link or node failure. Moreover, HighSSO has the feature to operate on heterogeneous WSN network

Further investigation of the proposed protocol will be the topic of future work including simulation and application to real life case (like effect of the shape of fire's front line Fig. 1). Additionally, simulation tool will be used to measure the effect of BS mobility on the recovery time in harsh environment. Furthermore, network traffic pattern and shape will be investigated. Also, the proposed hierarchical routing protocol will be enhanced to be tested under mobility of both BS and SNs. Finally, latest QoS routing protocols in WSN will be coded (using VC++ on OMNeT++ 3.3) and compared against HighSSO.

WSN involves more and more into our life. The importance is increased due to massive development on the hardware and communication abilities. These raise a demand from the clients (companies, governments and organizations) to provide a better or new solution for their applications. The target solution will save resources, human life, cut down the cost, get better management and provide new services. Still, the solution has not been reached completely, especially on the network layer with respect to the reliability and availability issues. For all of these reasons, WSN is one of the hottest research area and needs an enormous of research effort (as mentioned on the introduction).

# REFERENCES

1. MIT, *Emerging Technologies That Will Change the World*, in *IEEE Engineering Management Review*. February 2004. p. 20-20.
2. C. Siva Ram Murthy, B.S.M., *Ad Hoc Wireless Networks: Architectures and Protocols*. 1st Edition ed. May 24, 2004: Prentice Hall.
3. Romer, K.M., F., *The design space of wireless sensor networks*. IEEE Wireless Communications [see also IEEE Personal Communications], Dec. 2004. **11**(6): p. 54 – 61.
4. Akyildiz, I.F.W.S.S., Y.; Cayirci, E., *A survey on sensor networks*, in *IEEE Communications Magazine*. Aug. 2002. p. 102 – 114.
5. Al-Karaki, J.N., Kamal, A.E., *Routing techniques in wireless sensor networks: a survey*. IEEE Wireless Communications [see also IEEE Personal Communications], Dec. 2004 **11**(6): p. 6 – 28.
6. D. LAL, A.M., F. Herrmann, E. Uysal-Biyikoglu, A. Keshavarzian. *Measurement and Characterization of Link Quality Metrics in Energy Constrained Wireless Sensor Networks*. in *IEEE GLOBECOM*. 2003
7. G. Zhou, T.H., S. Krishnamurthy, J.A. Stankovic. *Impact of Radio Irregularity on Wireless Sensor Networks*. in *Proceeding of the Second International Conference on Mobile Systems, Applications, and Services*. 2004. New York: ACM Press.
8. Jean-Marie Gorce, R.Z., Herve Parvery, *Impact of Radio Link Unreliability on the Connectivity of Wireless Sensor Networks*. *Eurasip Journal on Wireless Communications and Networking*, v 2007, 2007. Article ID 19196.
9. Heinzelman, W.R.C., A.; Balakrishnan, H., *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*. in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000. 2000. Hawaii.
10. Heinzelman, W.B.C., A.P.; Balakrishnan, H., *An application-specific protocol architecture for wireless microsensor networks*. IEEE Transactions on Wireless Communications, 2002. **1**(4): p. 660 – 670.
11. Subramanian, L.K., R.H. *An architecture for building self-configurable systems*. in *First Annual Workshop on Mobile and Ad Hoc Networking and Computing*, 2000. *MobiHOC*. 2000.
12. Macedo, D.F., Correia, L.H.A., dos Santos, A.L., Loureiro, A.A.F., Nogueira, J.M.S., *A rule-based adaptive routing protocol for continuous data dissemination in WSNs*. *Journal of Parallel and Distributed Computing*, 2006. **66**(4): p. 542-555.
13. Ji, P., Wu, Chengdong, Zhang, Yunzhou, Jia, Zixi. *Research of Directed Spanning Tree Routing Protocol for Wireless Sensor Networks*. in *Proceeding of 2007 International Conference on Mechatronics and Automation (ICMA 2007)*. 2007.
14. Avresky, D.R., *Embedding and reconfiguration of spanning trees in faulty hypercubes*. IEEE Transactions on Parallel and Distributed Systems, 1999. **10**(3): p. 211 - 222.
15. Zhang Zhenjiang, L.y. *An Energy-Efficient Redundant Nodes Tree Mechanism for Wireless Sensor Networks*. in *Proceeding of 2006 International Conference on Systems and Networks Communication (ICSNC '06)*. 2006.
16. Moh, M., Dumont, M., Teng-Sheng Moh., *Evaluation of dynamic tree-based data gathering algorithms for wireless sensor networks*. *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, 2005: p. 170-175.
17. Tzung-Shi Chen, H.-W.T., Chih-Ping Chu. *Gathering-Load-Balanced Tree Protocol for Wireless Sensor Networks*. in *Proceeding of the 2006 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*. 2006.
18. Jichuan Zhao; Erdogan, A.T.A., T., *A novel application specific network protocol for wireless sensor networks*. IEEE International Symposium on Circuits and Systems, 2005. *ISCAS*, 2005. **6**: p. 5894 - 5897.
19. J. Li and P. Mohapatra, *An Analytical Model for the Energy Hole Problem in Many-to-One Sensor Networks*. *Proc. 62nd IEEE Vehicular Technology Conf. (VTC-Fall '05)*, 2005. p: 2721-2725.
20. [http://www.nifc.gov/fire\\_info/lg\\_fires.htm](http://www.nifc.gov/fire_info/lg_fires.htm)
21. VIJAY K. GARG, *Wireless Communications and Networking*, 1<sup>st</sup> edition, Morgan Kaufmann
- 22.C. Hartung, R. Han, C. Seielstad, et al., *FireWxNet: A multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments*. *Proc. of International Conference on Mobile systems, Applications and Services (MobiSys'06)*. Uppsala, Sweden, 2006, 28-41.
- 23.D. M. Doolin, and N.Sitar, *Wireless sensors for wildfire monitoring*. *Proc. of SPIE Symposium on Smart Structures and Materials*, San Diego, CA, 2005, 477-484.