

# Gradient based Temperature-aware routing in Body Area Sensor Networks

Ashwin Ganesh  
Krishnamurthy  
Department of Electrical  
Engineering and Computing  
Systems  
University of Cincinnati  
Cincinnati, OH 45220, USA  
krishnag@mail.uc.edu

Junghyun Jun  
Indian Institute of Technology  
Rupnagar, Punjab 140001,  
India  
peterjun@iitrpr.ac.in

Dharma P. Agrawal  
Department of Electrical  
Engineering and Computing  
Systems  
University of Cincinnati  
Cincinnati, OH 45220, USA  
dpa@cs.uc.edu

## ABSTRACT

One promising area of wireless body area networks is remotely diagnosing human health by implanting bio-medical sensors on or inside of a human body. Despite the benefit of such networks, one key challenge is that it should not cause harm at any time. Although the material of bio-medical sensors are made to be harmless, the heat produced from sensing and communicating between sensors can still harm the surrounding human tissues. Therefore, unlike traditional wireless sensor networks, any operations performed in this network must be aware of this issue.

In this paper, our goal is to route a packet along a cool path towards a coordinator node which is responsible for communication with the outside the world. Our approach is formulating this thermal routing problem as an artificial intelligence search problem. We use a temperature gradient as a cost function and select neighboring sensors that offer a cooler gradient value. Our approach is loosely based on the fringe search which avoids the overhead of sorting the nodes on every iteration of the search. We perform simulations and compare our proposed scheme with other existing well known schemes.

## Categories and Subject Descriptors

C.2.2 [Network Protocols]: Routing protocols; D.0 [Software]: General; I.6.0 [Simulation and Modeling]: General

## Keywords

Bio-medical sensors, Routing Algorithm, Body Area Network

## General Terms

Protocol

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

BODYNETS 2014, September 29-October 01, London, Great Britain

Copyright © 2014 ICST 978-1-63190-047-1

DOI 10.4108/icst.bodynets.2014.257002

## 1. INTRODUCTION

Body Area Networks (BANs) are considered as key building blocks for emerging applications of wearable technology as well as the *Internet of Things*. Starting from traditional network scenarios to today's advanced network infrastructure, BANs have played a significant role in developing the bio-medical sector. Presently, we have two types of BANs. One is an in-body area network [1] which implants sensors inside a human body and allows communication between invasive/implanted sensors. The other is on-body area network [1] which allows communication between non-invasive and wearable devices. The sensors used in both types of BANs cause some harmful effects on the surrounding body tissues and organs due to increase in the temperature of the sensors. While designing a routing protocol for in-body area networks, one must be aware of this operational temperature requirement unlike typical BANs. For example, invivo-sensor [2] is a special type of sensor that detects physiological changes inside the body environment. It senses the vital body data and transmits the information to a controller device, known as the *coordinator* node exploiting wireless communication. During the communication, the tissues inside the body absorb the radiation energy and convert it to heat [3]. Therefore, it is imperative that the operating temperature of sensors must remain low such that they do not harm the human being.

Energy is another issue related to BANs. In BAN, a direct communication between the invivo-sensors and the coordinator node is often not possible since the propagation loss around the human body is very high [4] and low transmission power of the radio is required to save the available energy in the sensors. In such a case multi-hop routing is used where, sensor nodes also relay data from their neighboring sensor nodes towards the coordinator node. In this network, there exist more than one path from the source sensor to the coordinator node. Unlike traditional routing protocol where paths are selected based on energy efficiency or maximizing throughput, we propose a thermal-aware routing approach. This approach optimizes the search for a cool path for packet transmission, thereby avoiding already heated up nodes.

Our approach is loosely based on *fringe search* [5] for path finding. We first define a cost function that is based on a temperature gradient and this is used to select the next forwarding node. Basically, forwarding nodes in cooler gradient areas are queued for immediate expansion, whereas,

forwarding nodes in hotter gradient areas need to pass a threshold condition to be expanded. All other nodes are expanded later (explained in section 3). Based on this iterative methodology, a path is selected to let packet generation take place.

## 2. RELATED WORK

Due to growing interest in WBAN and its harmful effect on human tissue, several temperature aware algorithms have recently been proposed. They are broadly divided into distributed routing and centralized routing.

Distributed routing requires exchange of local temperature of neighboring nodes and their hop distance to the destination node. Thermal-aware routing algorithm [6] (TARA) is a good example of this type. It basically collects hop count and the next hop for a particular destination from its own neighbors list. When packets need to be forwarded, it selects the coolest node in its neighbors list as the next hop. If it senses a *hot spot* in its path, the packet is buffered until the system cools down. The packet is discarded from buffer if it remains in excess of a time period. The hot spots are detected based on each node listening to its neighbor's activities (thereby indirectly assessing the temperature). If no activity takes place for a period of time, a sensor that has been marked as a hot spot, would be removed from the hot spot list as its temperature cools down to a threshold temperature. There maybe certain cases in which the a node cannot forward a packet to any of its neighbor as they are all in the hot spot region. In this case, a withdrawal strategy is used [3, 6]. The packet is sent back to the previous node. Whenever the system cools down and hot spot temperature falls below the threshold temperature the current node informs the previous node of this change. As the TARA protocol routes the packets away from the vicinity of the heated nodes, it adds a lot to the delay in packet arrival and thereby increasing the amount of data communication with the nodes. Least temperature routing algorithm (LTR) is another distributed routing algorithm. Similar to TARA, it forwards packets to its coolest neighbor. However, different from TARA, it forwards the packet to the second coolest node if the coolest node has received a packet recently. This way it prevents a cool node suddenly becoming a hot spot. Although both TARA and LTR uses hop count for selecting a forwarding node, it does not consider a short path towards the destination node. Adaptive least temperature routing protocol (ALTR) is an improvement over the LTR protocol wherein, a node which has generated a packet checks for the number hops to the destination node, if the number of hops is less than MAX\_HOPS\_ADAPTIVE, it follows the LTR algorithm otherwise it follows the shortest hop routing algorithm[3]. Hot spot Preventing Routing (HPR) [7] is another distributed approach which is seen as an improvement over LTR and ALTR. However, unlike LTR and ALTR, a hot spot is determined based on the average temperature of the neighboring nodes as well the temperature of the current node. It still suffers higher packet delivery delay as it drops packets when the number of hops go beyond the MAX\_HOPS threshold.

One problem in above mentioned distributed methods is that they often result in sub-optimal solutions. Since preventing sensors from harming human tissue is the most critical requirement, this sub-optimal solution is not sufficient for most cases. Takahashi et al[8] proposed least total route

temperature algorithm. This is a centralized algorithm in which the least temperature path is chosen from all possible routes. LTRT[8] approaches temperature-aware routing as a graph problem and calculates a single source shortest path (Using Dijkstra's algorithm etc.)for route calculation [8].

To formulate the thermal routing problem as a graph problem the protocol assigns the temperature of each sensor node as a weight of that node and during route calculation, the concept of transfer weight is used [8]. It can be seen that LTRT algorithm is a hybrid of SHR and LTR. While LTRT maintains the average temperature rise as a metric, there may be certain nodes in the network that have exceeded the safe temperature level and can be hazardous to human health [3].TSHR or Thermal-Aware Shortest Hop Routing [9] is an improvement over the HPR. It is very similar to HPR in both the setup and the routing phases. It consists of two thresholds a fixed threshold and a dynamic threshold [3, 9]. The dynamic threshold is used to mark a node as a hot spot and the fixed threshold is used to keep the temperature of the next hop node below a maximum temperature value. TSHR protocol enables lesser temperature rise as compared to HPR but a higher packet delivery delay since it does not employ any packet drop mechanism.

## 3. TEMPERATURE GRADIENT SEARCH

In this section we provide the details of our thermal-aware routing algorithm named as Temperature Gradient Search (TGS). TGS follows LTRT [8] in regards to handling the routing decision on an end to end basis. It constructs a coolest-path with minimum hops by using a *cost function*. Then, it considers the average temperature of the neighboring nodes like HPR and TSHR. Here, we assume the temperature of node and its surrounding region has been transmitted to their neighbors periodically in beacon signals.

1. A data exchange takes place between all the nodes in the network and each node stores its hop count to its desired destination node. This is known as the data-gathering phase.
2. The next step is the path generation phase. Where the source node calculates a  $C_{thresh}$  threshold value. This value is given by the following equation :
$$C_{thresh} = T_{node} + k * T_{avg}$$
 Here,  $C_{thresh}$  is the *optimistic threshold* value, which accepts a neighbor with a hotter temperature as long as the neighboring node is closer to the destination node as compared to the current best neighbor node.  $T_{avg}$  is the average temperature of a node's neighbors and  $k$  is the *normalizing coefficient* that ranges between 0-1. This coefficient decides how much of a factor the temperature of the neighboring nodes play. A very high value of  $k$  indicates a higher temperature neighbor being chosen for routing decision. A very low value of  $k$  indicates that the immediate next hop nodes are not factored for route calculation. The ideal value for this coefficient is 0.5.  $T_{max}$  represents a critical temperature that is just below the actual temperature that can cause damage to the tissue.
3. A single double-ended list  $L$  is maintained. For the nodes that have to be explored immediately, they are

added to the front of  $L$ . All other nodes are added to back of  $L$ . This is decided based on the  $C_{thresh}$  and the *temperature gradient* parameters explained below.

4. A route to the destination node is planned by iterating through  $L$ . Before expanding a sensor node's neighbors, the sensor node's temperature is always checked with the  $T_{max}$  value. If it does not fall below this value, this sensor node will be rejected and the next node in the fringe is queued for expansion.
5. When a sensor node's neighbors are explored, a cost between the sensor and its neighbors are calculated from the cost function below.

$$cost = diff\_temp(node_i, node_j) / distance(node_i, node_j)$$

Here  $node_i$  is the sensor node that is currently expanded and  $node_j$  is one of its neighbors. This equation calculates the change in temperature over a distance which is denoted as temperature gradient. At this juncture, the algorithm branches into three cases:

- (a) If a neighboring node's temperature gradient is a positive change or a cooler transition, if its neighbor's temperature gradient is higher than the current-best cost value then this neighbor is added to the front of the list and the current best cost value is updated with this neighbor's temperature gradient value.
  - (b) If a neighboring node's temperature gradient is not a cooler transition, but has a hop-count to the destination node better than the current-best neighbor node's hop-count to the destination node, as long as its temperature value is less than  $C_{thresh}$ , this neighbor is added to the front of  $L$ .
6. If the above conditions are not satisfied by the currently iterating neighbor, the neighboring node is added to the back of  $L$ .
  7. In any case, if the next hop node is the coordinator node (or destination node), this node is automatically chosen and the packet generation starts with the planned path.
  8. The planned path is obtained by using a parent pointer to backtrack from the destination to the source. If no path to the coordinator is found, the packet is dropped.

## 4. EXPERIMENTS

In this research we focus on the network temperature, nodal temperature and packet success rate as points of interest for determining which algorithm offers a better solution. We chose a mesh topology such that each node can send packets to any other node via multi hopping. In order to keep up with our research goal, we tested the following metrics - average temperature rise of the network, maximum temperature reached by any node in the network per simulation, the number of packets dropped per simulation and the percentage of hot spots passed by packet per simulation. We wrote a discrete event simulation program in Java. Our simulation model consists of data gathering, packet generation, route planning and packet transmit for a given topology. Instead of fixing one node as a *coordinator* and transmitting

packets from multiple source nodes, we generate a random source and a random destination node per iteration thus, effectively handling different types of node-node interactions<sup>1</sup>. We iterate for a total of 2000 time units and every iteration consists of one new packet being generated by the randomly selected source node. We vary the packet generation rate from 0.25 packets per time unit to 3.0 packets per time unit for all metrics. This kind of a model enables to identify how each algorithm performs from low to high traffic conditions and its effect on temperature and successful packet delivery. Each sensor node is given a base temperature of 1.0 before the start of the simulation. We assumed that the temperature of a node increases by 1.0 when it receives a packet and when it transmits a packet. We assumed a cooling value of 1.0 for every 40 iterations [8, 10]. This can be identified as the cooling of sensor nodes by convection in a real life scenario. We followed the simulation setup in [8, 10] for setting up of various parameters for the simulation of the other algorithms. For the ALTR algorithm the MAX\_HOPS\_ADAPTIVE value is used to change the algorithm to SHR if a certain number of hops have passed and the destination is not found. This value is set to 10 [8]. The normalizing coefficient  $k$  for TGS is set to 0.5 such that the influence of neighboring nodes is exactly half. In case of HPR, the MAX\_HOPS parameter is set 40[7], this means that once 40 hops have reached and no path is found, then that packet is dropped. For TSHR and TGS, the fixed threshold has to be set before the simulation. The fixed threshold value for both these algorithms are assumed to be 100<sup>2</sup>. To test the scalability of the algorithm we varied the number of sensor nodes from 20 (a 2x10 mesh topology) to 100 (a 10x10 mesh topology) and obtained similar results for all the metrics. We present the results of our simulations for a 5x10 mesh topology for which we calculated the aforementioned metrics.

### 4.1 Average temperature rise versus packet generation rate

Figure 1 represents the average temperature rise of the network with the packet generation rate. While ALTR focuses just on the node's temperature rather than the destination the packets tend to roam around the network causing a temperature rise. Also, when the threshold condition is reached, ALTR switches to SHR and hence routes packets through hot spots as well. LTRT uses the node's temperature as weights and uses Dijkstra's shortest path algorithm to get the coolest route to the destination. HPR tends to send the packet to the coolest next hop node in the shortest path, this leads to overall temperature increase of all the nodes in the network. TSHR uses a fixed threshold and hence the temperature rise is constant and could be said to be well within the norms. TGS always attempts to optimize the path as it uses temperature gradient and hop count. But, since the  $C_{thresh}$  condition accepts hotter temperature nodes, the overall network temperature does not rise much as other cooler nodes do not get heated up if a shorter path can be found. Furthermore the  $T_{max}$  parameter ensures that the temperature of the selected node is within the safety margin. So, TGS performs better than ALTR, LTRT and

<sup>1</sup>Sensor-Sensor and Sensor-Coordinator.

<sup>2</sup>This value does not reflect on the real world temperature values in Celsius or Fahrenheit.

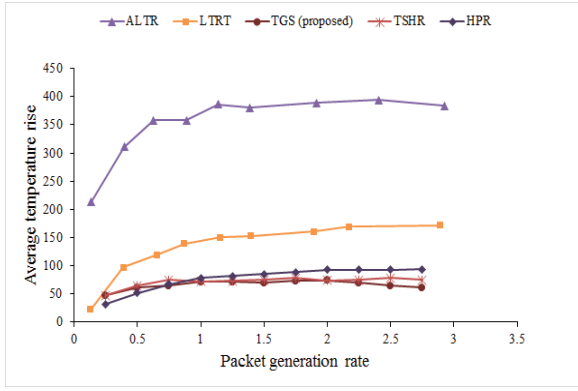


Figure 1: Average temperature rise versus packet generation rate. Simulation results of ALTR versus LTRT from Takahashi et al([8]).

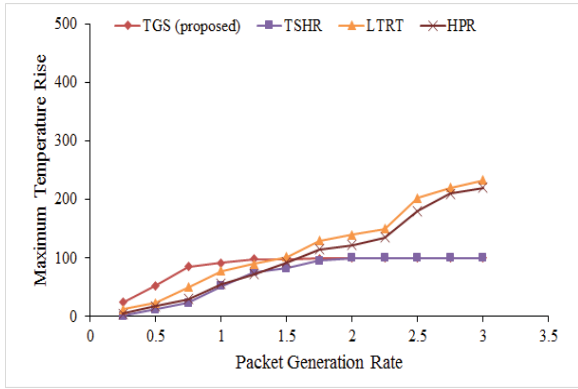


Figure 2: Maximum Temperature rise versus packet generation rate.

HPR and is on par with TSHR in this metric.

## 4.2 Maximum temperature rise versus packet generation rate

Figure 2 represents the maximum temperature rise versus the packet generation rate. For this setup, ALTR algorithm can be avoided as the average temperature rise for it was very high. For both TSHR and TGS, the maximum temperature rise does not go beyond their respective fixed threshold values. However, LTRT does indeed show a maximum temperature rise comparatively as only the overall path temperature is taken into account by this algorithm. While HPR has a dynamic threshold for hot spot detection, the maximum temperature rise is more than both TSHR and TGS algorithms as it routes the packets through the coolest next hop node and uniformly increases the temperature of all the available cool nodes.

## 4.3 Average hop count per packet versus packet generation rate

Figure 3 shows the average hop count per packet versus the packet generation rate. The hop count is defined as the minimum number of hops taken for a given source and a destination node. So, an average hop count is taken for

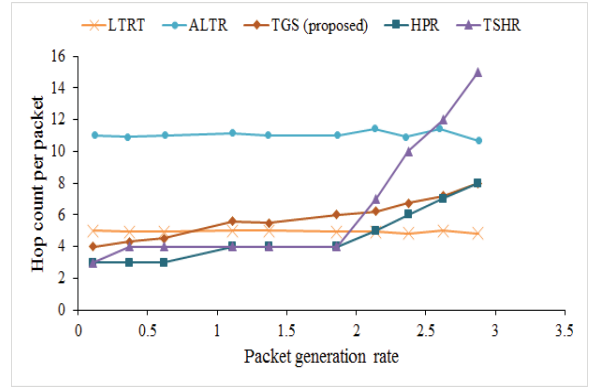


Figure 3: Average hop count per packet versus packet generation rate.

each iteration. The hop count metric can be useful for analyzing the time taken for a packet to reach a destination node. More the hop count, longer the packet takes. LTRT which provides an end-to-end solution maintains a constant average hop count value, while TSHR, HPR and TGS start with a low hop count value for smaller packet generation rates and they gradually increase. HPR is the best among the three as HPR uses the coolest next hop node for packet transmission and does not withdraw. However it drops the packet when the MAX\_HOPS parameter is exceeded. Similarly, if TGS is not able to find the destination node from the nodes available in the fringe, then packets are dropped. TSHR buffers the packets until a cool node becomes available. So, when the overall temperature of the system is very high and causes considerable delay in packet delivery rate.

## 4.4 Packets dropped versus packet generation rate

Figure 4 shows the number of packets dropped versus packet generation rate. As both ALTR and TSHR have no packet disregarding mechanism their packet dropped rate is 0. LTRT uses a routing decision based on shortest path and always finds a destination node. So, LTRT also has packet rate of 0. HPR on the other hand, drops packets when the number of hops exceed the MAX\_HOPS threshold. Initially, lesser number of hotspots are encountered and there is lesser detouring that takes place. For TGS, initially the nodes are cool so a lot of cooler gradients appear in the routing process. Once the packet generation increases the *optimistic threshold* comes into play and a neighboring node with lesser hop count to the destination takes precedence. But, since there is  $T_{max}$  threshold for node selection lots of neighbor nodes are not selected for routing and hence there is a considerable packet drop.

## 4.5 Percentage of hot spots passed versus packet generation rate

Figure 5 shows the percentage of hot spots passed versus packet generation rate. In our experimentation we assumed a hot spot to have a temperature greater than 100 units. As the average temperature rise of the network becomes very high for ALTR, the number of hot spots passed by packets in the ALTR algorithm, drastically increases as the packet gen-

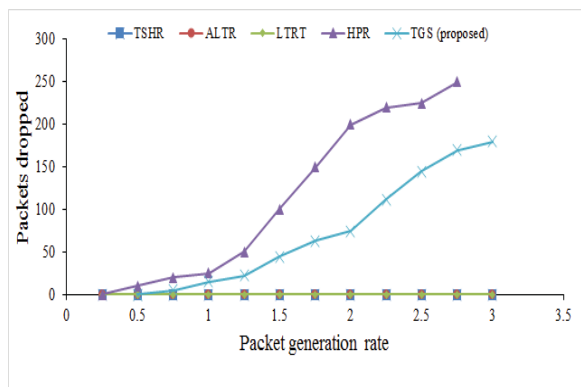


Figure 4: Packets dropped versus packet generation rate.

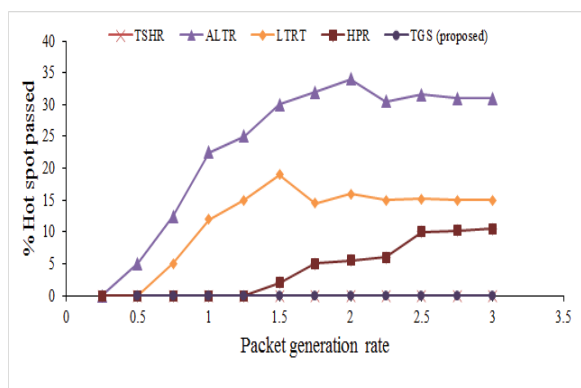


Figure 5: Percentage of hot spots passed versus packet generation rate

eration rate increases. LTRT produces less number of hops and lower average temperature rise (lesser than 150). But from our simulations, we observed as there is no threshold to particularly determine a hot spot for this algorithm and as a result packets under the LTRT algorithm pass through hot spots. HPR has zero hot spot passing till the packet generation rate increases. TGS and TSHR have zero hot spots being passed as the fixed threshold for each of these algorithms was assumed to be 100.

## 5. CONCLUSION

This paper presents thermal-aware routing approach for BANs. The proposed routing algorithm combines the end-to-end nature of the LTRT algorithm with the neighborhood based threshold calculation like the HPR and TSHR algorithms. Our routing algorithm keeps the temperature of the sensor node as the main priority. Therefore it can be observed from our simulation results that it maintains a low average network temperature, avoids hot spots, and has a very low maximum temperature rise compared to other algorithms. It also has a modest packet delivery delay as compared to other algorithms and a packet drop rate that increases with the packet generation rate. For our future work, we plan to run the proposed algorithm for different topologies and reduce the packet drop rate. We would also like to look at the energy consumption of this model and perhaps bring about

an efficient sleep cycle process associated with this model.

## 6. REFERENCES

- [1] Sana Ullah, Henry Higgins, Bart Braem, Benoit Latre, Chris Blondia, Ingrid Moerman, Shahnaz Saleem, Ziaur Rahman, and KyungSup Kwak. “a comprehensive survey of wireless body area networks”. volume 36, pages 1065–1094. Springer US, 2012.
- [2] Francesco Merli, LÁl’andre Bolomey, FranÁgois Gorostidi, Yann Barrandon, Eric Meurville, and AnjaK. Skrivervik. “in vitro and in vivo operation of a wireless body sensor node”. In KonstantinaS. Nikita, JamesC. Lin, DimitriosI. Fotiadis, and Maria-Teresa Arredondo Waldmeyer, editors, *Wireless Mobile Communication and Healthcare*, volume 83 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 103–110. Springer Berlin Heidelberg, 2012.
- [3] Christian Henry Wijaya Oey and Sangman Moh. “a survey on temperature-aware routing protocols in wireless body sensor networks”. *Sensors*, 13(8):9860–9877, 2013.
- [4] P. Honeine, F. Mourad, M. Kallas, H. Snoussi, H. Amoud, and C. Francis. “wireless sensor networks in biomedical: Body area networks”. In *Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on*, pages 388–391, May 2011.
- [5] Yngvi BjÁurnsson, Markus Enzenberger, Robert C. Holte, and Jonathan Schaeffer. “fringe search: beating a\* at pathfinding on game maps”. In *Proceedings of IEEE Symposium on Computational Intelligence and Games*, pages 125–132, 2005.
- [6] Q. Tang, N. Tummala, S. K. S. Gupta, and L. Schwiebert. “tara: thermal-aware routing algorithm for implanted sensor networks”. volume 3560, pages 206–217, June–July 2005.
- [7] Anirban Bag and Mostafa A. Bassiouni. “hotspot preventing routing algorithm for delay-sensitive applications of in vivo biomedical sensor networks”. *Information Fusion*, 9(3):389 – 398, 2008. Special Issue on Distributed Sensor Networks.
- [8] D. Takahashi, Yang Xiao, and Fei Hu. “ltrl: Least total-route temperature routing for embedded biomedical sensor networks”. In *Global Telecommunications Conference, 2007. GLOBECOM ’07. IEEE*, pages 641–645, Nov 2007.
- [9] M. Tabandeh, M. Jahed, F. Ahourai, and S. Moradi. “a thermal-aware shortest hop routing algorithm for in vivo biomedical sensor networks”. In *Information Technology: New Generations, 2009. ITNG ’09. Sixth International Conference on*, pages 1612–1613, April 2009.
- [10] A. Bag and M.A. Bassiouni. “energy efficient thermal aware routing algorithms for embedded biomedical sensor networks”. In *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, pages 604–609, Oct 2006.