XIX IMEKO World Congress
Fundamental and Applied Metrology
September 6−11, 2009, Lisbon, Portugal

A ROUTING PROTOCOL WITH DISTRIBUTED TOPOLOGY MAINTENANCE IN WIRELESS SENSOR NETWORKS

Andrzej Michalski [1,2], *Łukasz Makowski* [2]

[1] Military University of Technology, Warsaw, Poland, anmi@iem.pw.edu.pl
[2] Warsaw University of Technology, Warsaw, Poland, makowski@iem.pw.edu.pl

Abstract − In this paper we present some algorithms for devices with limited energy resources that are forming ad hoc wireless sensor network. Proposed Challenge-Answer MiniProtocol (CAMP) is a clustering and gradient-based protocol with all-to-sink and sink-to-all transmission style, derived from directed diffusion idea. It deals with challenges such as variable quantity of working nodes which are constituting a network, hot-spot problem and power resource constraints. Tests of network using CAMP are presented and discussed.

Keywords : wireless sensor networks, protocols, routing

1. INTRODUCTION

One of the most important problems in wireless sensors networks (WSN) is research for an algorithm that optimizes path between nodes and finds the best route. Ample solutions were developed to solve this issue although with constant progress in hardware further investigation in this area of measurement systems is considered to be necessary.

Both wireless and wired networks are usually represented as graphs where links are edges connecting nodes. Optimization of network connections is therefore a graph optimization in respect to some value e.g. total edges length or edges quantity [1]. The fact called “wireless multicast advantage” is heavily exploited in presented protocol. It is assumed that omnidirectional antennas are in use while all links are symmetrical end equal in maximum length.

2. PRINCIPLES OF EXISTING SOLUTIONS AND BACKGROUND

The CAMP protocol is built upon idea of directed diffusion [2]. There are many problems with diffusion such as blind flooding of network with redundant and unnecessary packets. It leads to a hot-spot problem, which is preventing useful communication with redundant data transfers. Second drawback of such an approach is that unnecessary transmissions discharge batteries much faster.

To deal with blind flooding disadvantages, most protocols employ variety of algorithms to establish ad hoc connections between nodes. When some network structure is built then it becomes possible to optimize packet routing. How to create a suitable network is a problem on itself. Promising approach is to create some gradient function related to nodes and therefore connect them hierarchically. Therefore transmitting the packet from sensor node towards a sink node is determined by some deterministic function. For example gradient can be an equation containing distance from a given point to the sink, measured in number of “hops” [3]. Such idea is also investigated and developed in this paper.

Even though the network is successfully established its topology can suffer from unexpected changes due to harsh environmental conditions or internal factors such as battery depletion. Existing algorithms such as LEACH rebuild their network periodically [4]. In CAMP there is no schedule for reconfiguring the system while instead of that, the network is maintained constantly. CAMP deals with some resource constraints of unmovable yet physically detached, remote devices, and is expandable in this aspect.

3. DEVICES AND THEIR COLOURING

Nodes of two basic types can be deployed in a measurement area: ordinary devices with capability of exchanging messages with their neighbourhood but without access to different levels of communication system nor ability to transmit data to far parts of network, a second group of nodes consists of sink devices that collect data from their clusters of simpler measurement nodes. Data can be transmitted to higher level of network (e.g. GSM) by the sink devices only.

There is an assumed guarantee that all nodes are distinguishable from each other thanks to some kind of unique address which can be inscribed during manufacturing process. This feature is necessary to correctly build up clusters with only one working sink node per cluster. Assumption of single sink per cluster actually does permit a use of redundant head nodes per subgraph and run some election algorithms such as in LEACH. However with discussed below capabilities that already exist in CAMP, such an algorithm seems to be unnecessary.

CAMP establishes gradient function to maintain network topology, called as colour which is a tuple **c:=(h,s)** of hue and saturation respectively. There is a set of **i** possible hue values $\mathbf{H_i}$, with one value $\mathbf{h_k}$ per each sink node **k**. Saturation is a measure of “strength” that is bounding the node to the sink which works as a head of the cluster.

Ordinary sensor nodes just after deployment to measurement area are not bounded to any cluster therefore have no saturation yet and are called “gray nodes”'. They will have to be marked with hue value unequivocally pointing their relation with one and only one sink at every moment. Sink device **k** spreads its colour by sending announcement messages containing a colour tuple **(h_k, s< 1,0)**. Any non-sink device which receives such a message compares the announced saturation value s_r with its own **s**.

In the first case s_r is higher than saturation value held by device, which convinces device that it should connect to a new cluster. Device changes its hue to a received h_k and its saturation to a received value of saturation s_r becoming part of possibly new forming cluster. In this case each non-sink device spreads information about its cluster.

In the second case a device can receive a message with s_r smaller or equal to its own **s**. Such message is simply dropped therefore there are no changes in relationship between the node and clusters.

In this manner clusters are formed with nodes sharing the same hue value h_k, where **k** should be the closest available sink node. Saturation function diminishes the further from its sink node the given node is, in term of quantity of hops. On the Fig.1. there are shown established saturations after some time of protocol working with 90 simple nodes and 10 sink nodes. The values from random nodes positions are mapped to a mesh to create the graph. As the nodes were randomly scattered around centre point that is the maximum of the graph, which is the space around which connections and clusters have the strongest relations. The lower the value on the graph is, the longer path from node to cluster head.

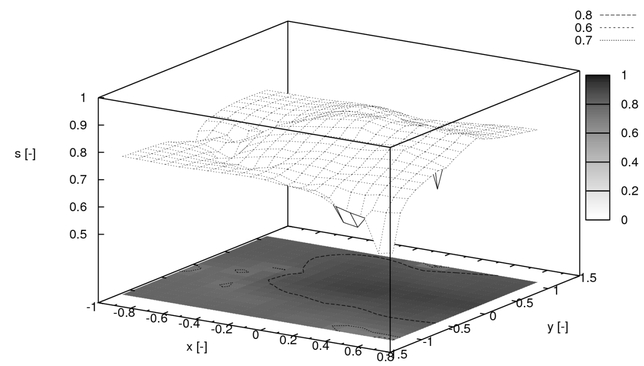

Fig. 1. Saturation gradient established with CAMP.

The clustering in CAMP is important, although it is only a basis for transmission which consists of four types of packets.

4. PACKET TYPES, TRANSMISSION AND PROCESSING

Each node has a queue of messages that should be sent. There are four types of packets:

- BID – announcing node existence,
- SEND – containing measured values,
- ACK – acknowledging packet retrieval,
- CALL – re-establishing lost connections.

SEND and CALL are messages that are sent towards sink node while ACK and BID move in opposite direction. It was mentioned in introduction that actually wireless signal has no specific direction but is available in some area and all nearby nodes can receive it. Therefore each node must decide if it is accidentally overhearing the packet which should be ignored or if it actually have to process an incoming data. Dealing with packets will be discussed now.

Approach to the test implementation was based on object oriented design and programming. Moreover, as for now there are not any concurrent implementations, so a problem of possible incompatibility is not existing yet. Therefore it was left to be decided in the future how to exactly serialize objects of CAMP messages to stream of bytes transmitted by physical links. However, in the paper there are proposed sample, serialized versions of all four CAMP packets.

4.1. Packet BID

The BID message is the most important one during the early stage of network life and when connections are being built. The packet must have fields such as:

1. packet type identifier,
2. tuple of proposed colour,
3. address of sending node,
4. sending node bid value.

Colour tuple should be checked against the rules described in previous part of the paper. However it would not be enough to correctly manage a network with more than one sink node. Possible cases of comparisons between set and received hue and saturation values are shown in Table 1 and numbered cases are discussed below.

Table 1. Cases in processing of a received colour

	$s_r < s$	$s_r = s$	$s_r > s$
$h_r = h$	1 – drop	2 – add to parent list	3 – change
h is “gray”	1 – drop	1 – drop	3 – change
$h_r \neq h$	1 – drop	1 – drop	4 – consider change

After processing the BID packet that enforced change of node colour, the node informs its neighbourhood about the change in cluster structure. Saturation that is being sent always have to be smaller than the value held by the node. Formula to calculate value of saturation that will be sent is not important as long as the produced gradient is strictly monotonic and message passing algorithm follows correct directions.

Usually when packet contains saturation not greater than actual saturation of the receiving node, it is dropped. This case is numbered as the first in the Tab.1.

There is one exception to the above rule, which is numbered as the second in the Tab. 1. In this case a path from receiver through sender to the sink node is as long as all other already discovered paths. Therefore receiver only updates its parent list with information about the possible route. With such a list maintained locally by each node it is possible to flexibly handle a tree, which is rooted in the sink node, in a distributed way.

In third case the node changes its whole colour when it was not connected to any cluster or just a saturation value when it advances in the rank of its current cluster. In both of these situations an empty parent list is created. Node appends to it an information about the node which transmitted the processed BID packet: its address and bid value. If there is any previous list of parent nodes, it is destroyed as invalid. Each node locally maintains its own list which is not exchanged nor transmitted between devices.

In the last – fourth case, we consider possible situation that node, which is already bound to a cluster, hears a BID packet from a different cluster. Simple difference between saturation proposed and set is not enough to decide if the node actually should change its colour tuple. The node can be a crucial router in its cluster therefore switch to a different one could imply ample changes to more than this one node. Moreover after simple switching it could again receive another BID packet from its former cluster which would lead to a jitter of nodes colour and unnecessary waste of energy. Therefore in such cases some hysteresis must be used in switching function. So then in the decision process a third value that is called a bid value is considered.

The bid value **b** is a calculated summary of node will to work as a network router. To evaluate this number a formula is proposed:

$$b = \frac{1 - l_{avg}}{k} \times \sum_{k} f_k \quad (1)$$

In the (1) all values except **k** are normalized to **<0;1>**. A group of similar factors $\mathbf{f_k}$ is taken on average. This number is then multiplied by an average idle time of device CPU, which is considered to be an independent factor. Average idle time is complementary to 1 in respect of time that CPU is being busy, therefore the equation is provided with the difference of them. In presented simulation only momentary battery fill and average load were used, but there could be more factors (e.g. insolation of device solar cells). Of course the list of parent nodes becomes deprecated shortly as the numbers taken into calculations varies constantly but this problem is dealt with elsewhere. Nevertheless having the bid value, the device can choose a better route between those that were formerly alike.

Before producing new BID message a node clears up its unsent messages queue of BID messages as they are deprecated. In other words there can be only one BID message scheduled per node and the new message takes place of the older one in the queue.

4.2. Packet SEND

This packet carries actual measurement data towards the cluster head. It contains fields such as:

1. packet type identifier,
2. address of sending node,
3. sending node colour tuple,
4. receiver address,
5. receiver bid value taken from list of parents,
6. address of node which performed measurement,
7. payload which contains measurement data.

This packet is directly addressed to a given node from the list of parents held by sending device. If the receiver is not a cluster head it should send the message further. But prior to that it have to provide new values to fields 2-5, current with respect to its own position. Of course fields 1, 6 and 7 should not be touched.

In general each SEND packet must be acknowledge by its receiver with an ACK packet, which is discussed below. After transmitting the SEND packet a node moves it from ordinary queue to a queue of SEND messages waiting for acknowledgement. If the message is not acknowledged in assumed time, it is being sent again.

To avoid possible faults while transmitting SEND messages, the origin of SEND message, which is the node that provided the measured value, calculates an unique ID number of the message. Along with acknowledging the SEND packet, the receiver saves the ID. If a node detects that it received the same SEND packet again, it does not sent it further although it should send a correct acknowledgement. Of course it would be pointless and hardly achievable to hold IDs of all messages that passed through the node. However, even short FIFO queue significantly prevents duplication of messages.

Different problem is that the sending node could have chosen suboptimal route. In this case it is highly probable that the message is overheard by a node which considers itself as a better route. It happens when the overhearing node has better saturation than indicated by message or at least the same saturation with higher bid value. In both cases the node announces its capabilities with BID packet. It is hoped that neighbouring nodes will update its parents lists and consider new route in the future. However, to lower the interference with acknowledging procedure, which should take place as quick as possible, sending of BID packet in this situation should be delayed.

4.3. Packet ACK

ACK packet is strictly related to SEND packet and some of its ideas are covered above. ACK packet must have fields such as:

1. packet type identifier,
2. ID of the SEND packet that is considered to be acknowledged with the ACK message,
3. address of proper receiver.

The ACK packet must be sent immediately after receiving the SEND message. Processing of this message is an easy procedure of removing from queues the SEND message that is unacknowledged yet.

4.4. Packet CALL

Parent node might get physically damaged, or may turn off after battery discharge, or finally may just change its rank in the cluster without successful notifying its children nodes. In other case the child node should strip the parent list on its own if there had been few ineffective communication attempts with a particular receiving device. In extreme repetition of this situation can lead to loosing all paths toward the sink node. To deal with the problem, the node broadcasts a CALL packet.

This is probably the simplest of all CAMP packets, which must contain only few fields:

1. packet type identifier,
2. address of sending node.

Any node that receive a CALL packet should broadcast a BID message according to rules described above.

5. IMPLEMENTATION AND SIMULATION ANALYSIS

A dedicated software in Python scripting language was written to:

1. test correctness of CAMP design,
2. check its actual properties and features,
3. measure the characteristics such as reliability and efficiency.

All data structures and procedures constitute a class *node()*. The class inherits some procedures from a *message()* class designed to hold and maintain message queues and queue items. These classes after minor improvements work also in real devices.

5.1. Investigation of basic characteristics

In the first simulation test it was investigated a network of nodes concentrated on plane around one point as in Gaussian distribution. Such placement of nodes can happen for example when devices are deployed from one source such as an aircraft.

The reliability of the CAMP protocol can be considered as ratio of SEND messages successfully received by sink nodes to all such messages, which have been produced due to measurements. From the graph that is presented on Fig. 2., it is possible to conclude about good scalability of network which is using CAMP. With increasing number of simple nodes and constant quantity of the sinks, amount of successfully transmitted messages drops. Due to collisions more messages are lost as it is expected. Increase of both simple and sink nodes as well, sustains message delivery probability. Therefore while proportion of simple to sink nodes is being kept constant, the considered measure for reliability is also firm.

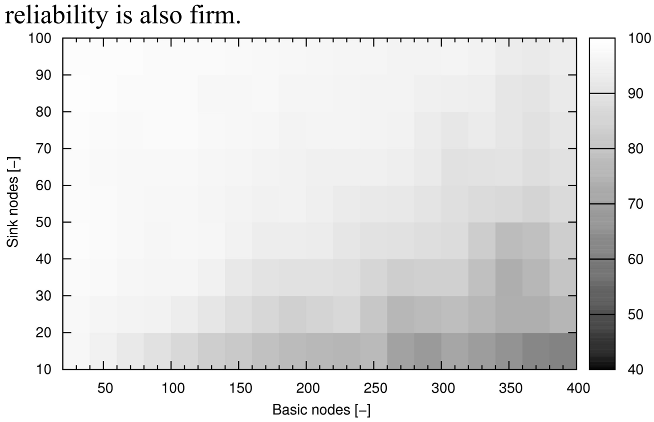

Fig. 2. CAMP message delivery reliability [%].

In the introduction it was said that CAMP protocol is based on directed diffusion idea. One important drawback of this kind of algorithms is the problem of duplicated messages. In the previous section it was stated that presented approach deals with this problem with small and simple queues held by each node. What leads to this issue is an unsuccessful ACK procedure. It happens when node repeats SEND transmission but within different cluster. As clusters are maintained separately then both can deliver the same message to two different sinks.

With the simulation software it was possible to investigate the problem further. On Fig. 3. it is presented an average ratio of messages that were delivered successfully but redundantly. Important conclusion is that waste of bandwidth due to this problem is insignificant. It was observed that in worst case no more than 1% messages were multiplied.

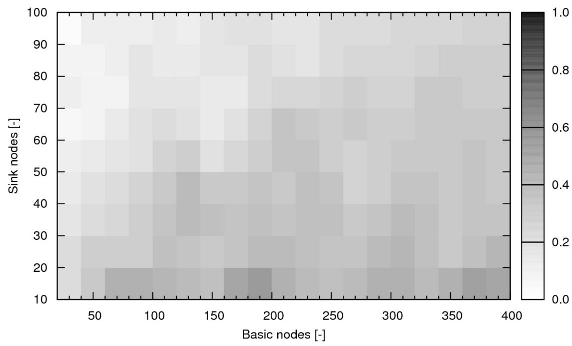

Fig. 3. Problem of redundant messages [%].

The queue, which helps to prevent multiplying the messages, in authors implementation of the protocol provides space for 64 packets. Therefore, with unique ID of length 16B, data structure occupies only 1kB. In authors opinion it is not much comparing to achieved benefits.

5.2. Test of self-optimization and self-repair

Second of the simulations was designed to focus on problem network topology optimization, which is happening throughout the network lifetime. There is no need to re-establish all connections if most are correct. So this procedure constantly improves connections within local scope.

Network topology chosen for this test was based on accurate rectangular mesh with distances between the closest nodes equal to 0.8 R_r where R_r is maximum range of their symmetrical communication links. To the accurate positions of devices was added an uncertainty equal to 0.25 of distance between the closest nodes. According to this, distances between nodes are in range from 0.4 R_r to 1.2 R_r. The number of devices in this experiment equals 100 where 4 of them are sink nodes.

Sample network of this kind in early stage of development is shown on Fig. 4. Thick lines are connections that were already discovered in ad hoc manner while thin are possible ones. Four black dots in four quarters of mesh are sinks. Gray background disk is place of interesting events. Sink node in lower left part started early and its cluster occupied too large part the network which is suboptimal topology.

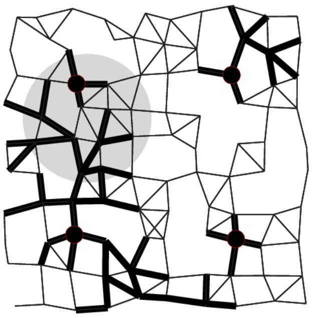

Fig. 4. Suboptimal topology in an early stage of mesh network.

After some time, which in presented case was no longer than 25 s, in some places it is observable how CAMP protocol is optimizing the network. On Fig. 5. the most interesting optimization is on the left, marked with gray background disk, where the whole branch of nodes switched to a different cluster. Distances between nodes and sink of their new group shortened thanks to this event, which makes further communications more efficient.

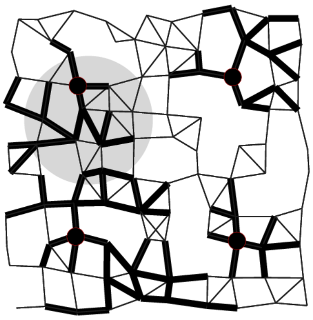

Fig. 5. Optimization at later stage of mesh network.

Obviously more changes are possible to improve network structure and actually such a process is ongoing all the time as the new nodes may connect or disappear from the sensors network based on CAMP. The real problem is when such a damaged node was the working sink.

Situation where sink node is abruptly turned off or suddenly damaged, makes the whole cluster unusable. However, due to CAMP feature which constantly improves the topology, it is expected that even such serious condition can be dealt with.

In the third simulation experiments we used a set of nodes places in a double ring. There are two nodes symmetrically placed on edges of ring as it is shown on Fig. 6. Such network is hard to optimize due to limited average number of connections per node. Damaging some nodes makes the ring even thinner that is with the diameter of only one node within the event area.

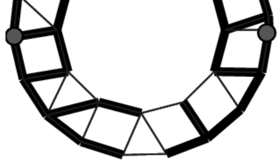

Fig. 6. A sample of hard to maintain ring network before damage.

After both clusters were established with all nodes adherent to any of them, simulation software turned off one of the sinks. The other cluster, which was still operational, successfully took over all remaining nodes. It took relatively long time of almost 1/6 of total network lifetime to repair the connections but what is important is that it finally happened. Network after the recovery from this disaster is shown on Fig. 7.

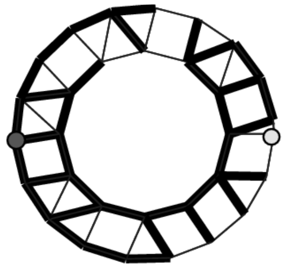

Fig. 7. The ring network after reconstruction.

On the basis of presented simulations authors claim that CAMP protocol is suitable do be deployed on real hardware. First attempt was supported with commercial modules. However, these devices have advanced features such as clustering and addressing single, particular device, only simple access to physical media was used in the experiment. Therefore CAMP algorithms had to deal with frame collisions. This experiment confirmed qualities which were found during simulations.

6. CONCLUSIONS AND FUTURE GOALS

The proposes protocol is simple, easy to implement and not very greedy for resources. It consolidates several layers of classical OSI model. CAMP is proved to be working correctly in simulated environment with very good scalability and with use of radio transmission provided by IEEE 802.15.4 physical layer.

Object oriented approach in developing the protocol opened a way for an easy move from simulations to actual sensor network.

Despite the fact that CAMP was designed for unmovable devices, its possible adaptability to MANETs is yet to be determined in future work. Authors consider it as an obtainable aim because of distributed manner in which the network graph is managed.

Main drawback of the design is that there is no packet fragmentation considered. Such a feature would greatly complicate its plain design and imply necessity of features that are unavailable in typical, simple devices of WSN. It must be assumed that transmitted measurement data is limited in size depending on physical network layer capabilities.

Finally the proposed gradient algorithm creates clusters with similar lengths of paths. This leads to a situation where clusters differ in size measured in quantity of nodes. It is yet to think of a correct balance between lengths of paths and number of nodes per a cluster, and influence of this dependence on network parameters such as maximum lifetime or average throughput.

In conclusion it is believed that presented protocol is suitable for systems with typical WSN constrains. Although relatively simple it is a reliably working solution, according to conducted experiments. It is believed that further development of CAMP can bring advancements in the field of wireless measurement systems.

REFERENCES

[1] R. Krishnan, D. Starobinski, "Efficient clustering algorithms for self-organizing wireless sensor networks", *Ad Hoc Networks*, vol. 4, pp. 36-59, 2006.

[2] C. Intanagonwiwat, R. Govindan, D. Estrin, and J. Heidemann, "Directed diffusion for wireless sensor networking", *IEEE/ACM Trans. Netw.,* vol. 11, pp. 2-16, 2003.

[3] C. Karlof, Y. Li, and J. Polastre, "Arrive: Algorithm for robust routing in volatile environments", University of California, CA, Tech. Rep. USB//CSD-03-1233, Mar. 2003.

[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy- efficient communication protocol for wireless microsensor networks," *Proceedings of the 33rd Hawaii International Conference on System Sciences - 2000*, vol. 2, p. 10, 2000.