RollingBall: Energy and QoS Aware Protocol for Wireless Sensor Networks*

Roman Špánek

Institute of Computer Science,
Academy of Sciences of the Czech Republic,
Pod Vodarenskou vezi 2,
Czech Republic,
spanek@cs.cas.cz;
Technical University of Liberec,
Halkova 6,
Czech Republic,
roman.spanek@vslib.cz

Abstract. In the paper, we present a quality of service and energy aware communication protocol, called RollingBall. We do believe that QoS and energy awareness are two of the most important parameters in wireless sensor networks. The protocol is completely distributed with no centralized control. The key idea is to introduce a resistance calculation for every connection in the network. The resistance reflects the distance to the sink together with energy capabilities of particular sensor. While the resistance is continually re-calculated, packets are sent to the sink via an appropriate path. Such a scheme allows to spend minimum messages on network management, whereby sensor network lifetime is extended and throughput remains high.

1 Introduction

The resent progress in micro-sensor technology, low-power analog/digital electronics devices and wireless-technology have led to the development of micro sensor networks. A sensor is generally equipped with a sensing unit, a radio transmitter, a processing unit and a battery providing electric energy for the whole sensor. The sensing unit provides measurement of the sensor surrounding and transformation of such measurements into an electric signal. The measured data are processed by the processing unit and then sent via the radio transmitter to a command center (sink). The sink is responsible for forwarding measured data into common networks (for example TCP/IP).
During past years there have been many projects addressing data gathering,

* The work was supported by the project 1ET100300419 of the Program Information Society (of the Thematic Program II of the National Research Program of the Czech Republic) “Intelligent Models, Algorithms, Methods and Tools for the Semantic Web Realization” and partly by the Institutional Research Plan AV0Z10300504 “Computer Science for the Information Society: Models, Algorithms, Applications”.

data processing and transmitting.
Although sensor networks are similar to ad-hoc networks, there exist some limitations tightly connected with sensor networks that cannot be found in ad-hoc networks. Such limitations can be summarized as follows:

- In wireless sensor networks it is impossible, in general, to build a centralized addressing scheme for possibly huge amount of deployed sensors. Classical network addressing schemes, like IP-based protocols, cannot be used.
- Due to usually large amount of sensors, often densely deployed, data generated by sensors suffer by high redundancy. Such a redundancy and the corresponding communication overhead ought to be reduced by a protocol.
- The other constrains are connected with the physical limitation of tiny sensors (e.g. transmission power, on-board energy, processing and storage capacity).

All this has led to the design of many new algorithms addressing the problem of routing data in sensor networks taking into consideration their special characteristics. The protocols can be generally classified as data-centric, hierarchical and location-based. The data-centric protocols are query-based protocols using data naming. Clustering techniques are adopted by the hierarchical protocols and position of sensors is utilized by location-based protocols. The last category includes network-flow and quality of service (QoS) aware protocols.
Sensor networks are able to cover broad spectrum of applications. For example, disaster monitoring and prevention, sensing in danger areas and environments and also many military applications.
The rest of the paper is organized as follows. The next section goes briefly through the state of the art and section 3 describes the aims of the protocol. Section 4 describes our algorithm in details and section 5 provides some discussion the protocol. The last section concludes the paper.

2 State of the Art

As was mentioned above, many protocols have been designed to fit all restrictions posed on sensor networks. The oldest and also simplest one data-centric mechanisms are flooding and gossiping [1]. Both mechanisms are devoted to data delivery without need for any routing algorithm. On the other hand, both suffer from communication overhead, despite the fact that the gossiping can do some improvements. Other data-centric protocols solve the problem of communication cost by employing a type of data announcement. As an example can be mentioned Sensor Protocols for Information via Negotiation (SPIN) [2], where the sensors in the network send meta-data describing measured data available at particular sensor. Such information is afterwards used for data delivery. Another protocol worth mentioning is Directed Diffusion [3] [4], where the sink sends its interest in the form of attribute-value pairs. Each sensor stores such advertisement in its cache for further use. On sensing demanded data, every sensor knows the cheapest path discovered during the advertisement phase to

the sink and sends along it the data. There are other protocols worth noticing that can be found in [5].
Hierarchical protocols are highly efficient in huge sensor networks and they are mainly based on sensors clustering. For every cluster is elected a leader responsible for communication support to the sink. Being particular, Low-Energy Adaptive Clustering Hierarchy (LEATCH) [6], Power-Efficient GAthering in Sensor Information Systems (PEGASIS) [7] and Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [8] are good examples of hierarchical protocols.
Protocols addressing QoS in sensor networks are very important. Namely, Energy-Aware QoS Routing Protocol, which was proposed by Akkaya and Younis [9], classifies data on real-time and non-real-time. Such classification helps forwarding real-time data, obviously more prone to delay, in the shortest possible time.
Although the protocols have improved sensor network lifetime (defined as the time till the first sensor is drained of energy) significantly and QoS task has been also advanced, there is further space for improvements. We consider the network lifetime and QoS as the most important parameters and therefore we address these parameters in our approach, described in the following.

3 Design Aims

In this section we formulate the aims of our protocol more precisely. The aims of our approach are:

- to minimize communication spent on the network management
- to improve QoS while maximizing the network lifetime

As these requirements are not easy to implement, according to our knowledge, they are usually considered separately.
For lifetime maximization it is crucial to have completely distributed algorithm with no centralized control mechanism spending sensors' energy on its management. Therefore our approach requires no centralized controls.
The QoS improvement can be achieved through maintaining some information about the shortest path to the sink. Note that the shortest path might be in terms of communication delay, hops needed to reach the sink and so on.
Our approach maintains information about the shortest path in terms of a gradient. In other words, our aim is to model an inclined plane acting as sensor network and a ball rolling on it as data. The inclined plane will be under continual evolution during the network lifetime reflexing amount of energy available at each sensor. Our protocol, called RollingBall, achieving such demands is described in the following.

4 Algorithm

We assume a network consisting of n sensors 1,2,...,n randomly distributed over a region and a sink labeled s. The whole network can be modeled as a weighted

graph with vertices representing sensors and edges representing connections between adjacent vertexes. The edges weights might represent communication cost, throughput or energy needed to transmit over the connection (precise definitions are given in the next section (4.1)).
In our approach we will further assume that the sink and sensors cannot change position. Further we will assume that communication is distance dependent and therefore resistance respects it.

4.1 RollingBall Phases

RollingBall algorithm can be divided into four stages.

1. neighbors' set evaluation
2. the gradient calculation
3. resources distribution
4. resources re-distribution

Neighbors' Set During the first stage two main calculations triggered on sensor deployment are made; the gradient determination and neighbors' set creation. The latter is done through scanning sensor's neighborhood and storing signal strength for each node in its vicinity. Each sensor also calculates its uniform *energy packet* as:

$$E_i = Energy_i/(N_i * \pi) \tag{1}$$

where $Energy_i$ is total amount of energy available at sensor i,
N_i is amount of sensors in its neighbors' set, and
π is given number determining how many times re-distribution phase can be triggered.

The Gradient Calculation The gradient calculation is given by the following procedure. The sink sends a *gradient message* to all sensors in its neighbors' set with its weight set to 0 by default. On receiving the message, a sensor derives its weight (based on the distance between it and the sender, measured by signal strength).
Then the sensor calculates the gradient (2) as the difference between sender's and its weight and forwards the message, now with updated weight attribute in it, into its neighbors' set.

$$G_{i,j} = (W_j - W_i) \tag{2}$$

As weights are continuously increased, the gradient will always point to the sink. Efficient network broadcast based on the search tree ideas used in parallel computing protocols like MPI, will be used to reduce communication overhead. Example situation is depicted in Figure 1.

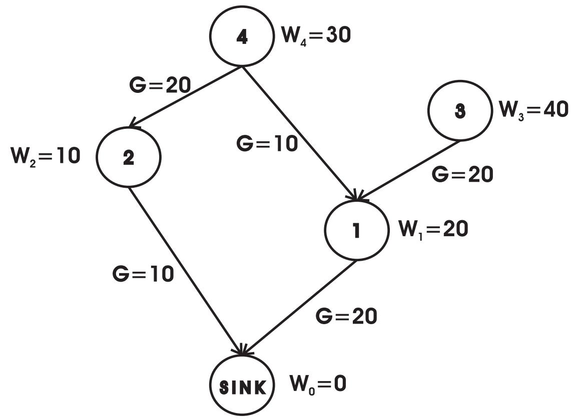

Fig. 1. The gradient calculation. Each sensor stores its weight calculated with respect to distance and gradient is subsequently derived as difference between their weights

For example, the gradient between sensors 1 and 3 is given by $W_3 - W_1 = 20$. This gradient is than assigned to edge 1-3.

Once the gradient has been calculated and stored by every sensor, *resistance* for each connection with neighbors is inferred as:

$$Res_{i,j} = 1/(Speed_{i,j} + G_{i,j}) + Round \tag{3}$$

where $Speed_{i,j}$ is the bandwidth for edge $i - j$,
Round is used later during re-distribution and it is set to 0 during this stage.

Resources Distribution During second, the resources distribution stage, each sensor in the network sends into its neighbors' set amount of energy packets E_i that can be send through it. In other words, a sensor sends to each its neighbor amount of packets that can be transmitted over it. Note that not all the available packets have been distributed, more over only a small part of the total amount has been distributed (depends on π value in (1)). The rest is then re-distributed in the next stage (see next section).

Resources Re-distribution The resource re-distribution phase is responsible for distributing energy packets when needed by any of adjacent sensors. Such situation occurs if any of adjacent nodes has used all given energy packets up. Note that both nodes know that such situation arises because both nodes have information how many packets had been distributed and also how many packets have been sent (see table 1 for example). Two main cases can occur. First, the sensor hasn't distributed all its energy packets E_i (1) and therefore more can be granted ($Round < \pi$). In case of packets available at the node, the node has

to recalculate resistance for the edge with the node demanding energy packets with increased *Round* value by equation (3). Although the amount of packets that can be sent over sensor i by sensor j is now reset, the edge resistance has also been increased. Therefore the edge is more expensive and packet delivered through has to have a higher $packet\ force$ (see 4.3) value.
As the second case that can occur during this phase of RollingBall is a case where there is no available energy to be granted ($Round = \pi$). In this case sensor demanding additional energy packets hasn't received any response in a predefined time threshold therefore it knows that the sensor asked for additional energy packets is exhausted and cannot be used for packets delivery in the future. Thence, with no additional transmissions every node in the structure is given by knowledge about status of all its neighbors.
Table 1 shows which information is stored by a particular sensor in the network.

Table 1. Sensor storage requirements for sensor 4 in Fig. 1

sensor No.	gradient	resistance	$E_i^G - E_i^{GS}$	$E_i^T - E_i^{ST}$
1	10	10	3	5
2	10	50	1	7
Round				

Particularly, sensor marked 4 in Fig 1 stores information for each of its neighbors (sensors 1 and 2). Each row in table 1 represents the following: a number of sensor in neighbors' set, gradient and resistance given for connection with the neighbor. Value $E_i^G - E_i^{GS}$ represents difference between amount of energy packets sent by sensor 1 over sensor 4 and amount of energy packets given to sensor 1 during distribution or re-distribution phase. In other words, how many energy packets can be sent by sensor 1 through sensor 4. Value $E_i^T - E_i^{ST}$, on the other hand, stands for difference between amount of energy packets transmitted by sensor 4 over sensor 1 and amount of energy packets that can be sent by sensor 4 over sensor 1. In other words, how many packets can be sent by sensor 4 through sensor 1 till all given energy packets are spent. The *Round* is incremented on each re-distribution phase and then used for resistance (3) re-evaluation.

Although the first and the second phases require some energy to be spent on management (e.g. discovering adjacent sensors), it is triggered only once during network lifetime. During the rest of lifetime only little energy is spent on energy packets re-distribution with minimal communication cost while both interested sensors know the time of depletion. We believe that this strategy can improve network lifetime significantly.

4.2 Energy Aware Layer

The energy aware layer is responsible for maximization of network life time while throughput remains high as possible (see section 4.3). The aim of the layer is to divert packets to a path with the minimal resistance. While the resistance is under continual evolution during the re-distribution phase, paths used for packets transmission change as well. The path is chosen on the following value:

$$Path = \min_i(Res_{i,j} - PacketForce) \quad (4)$$

where $PacketForce$ is an integer number assigned to every packet by a sender (see 4.3).
Therefore the layer distributes packets along different paths and exhausts them uniformly. Note that this calculation is performed by sensors on delivery path only. This approach is similar to a ball rolling on an inclined plane. The ball will follow the steepest descent. The same behavior is adopted in our approach.

4.3 Quality of Services aware layer

Since the precedent section was dedicated to the energy aware aspect of RollingBall, this one considers QoS is sensor networks. QoS aware protocols consider end-to-end delay and try to minimize it.
In our solution every packet is given by a $PacketForce$ integer number that reflects whether the packet is real-time or non-real-time data. The real-time data are given by a higher value then non-real-time ones. While path selection (4) depends on the $PacketForce$ and also an edge resistance, the real-time data have bigger change to use the shortest path (along the gradient) then non-real-time data. Due to this, the non-real-time data are delivered by sensors with lower traffic and therefore more available energy.
We believe that such scheme can significantly improve the sensor network lifetime and also due to preferring real-time traffic, real-time packets are delivered quickly along the shortest path (the largest gradient).

5 Discussion

I'd like to base an application to Fulbtight stipend ship on the paper. Therefore the prove would be the aim of the project for Fublright organization if appropriate.

6 Conclusion

The paper presents a new approach for treating QoS and energy issues in wireless sensor networks. Although many excellent protocols have been proposed there are only few addressing both QoS and the network lifetime.
RollingBall solves the problem of efficient packets delivery and also energy awareness by adopting a weighted graph model of sensor network. Every edge (connection) in the model is given by a resistance value that is calculated with respect to a gradient and energy resources available at particular node. The gradient points to the sink and is further used for QoS improvement. The resistance is then under continual evolution during network lifetime and packets are therefore

forced to be transferred along the most appropriate path to the sink while saving as much energy as possible. The accent is also put on spending very little energy on network management.

References

1. S. Hedetniemi and A. Liestman, "A survey of gissiping and broadcasting in communication networks" in *Networks*, vol. 18, No. 4, pp. 319-349, 1988.
2. W. Heinzelman, J. Kulik and H. Balakrishnan, "Adaptive protocols for information dissemination in wireless sensor networks" in the *Proceedings of the* 5^{th} *Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, August, 1999.
3. C. Intanagonwiwat, R. Govindan and D. Estin, "Directed diffusion : A scalable and robust communication paradigm for sensor networks" in the *Proceedings of the* 6^{th} *Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00)*, Boston, MA, August, 2000.
4. D. Estrin, et al., "Next century challenges: Scalable Coordination in Sensor Networks" in the *Proceedings of the* 5^{th} *Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, Seattle, WA, August, 1999.
5. K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks" in the *Elsevier Ad Hoc Network Journal*, Vol 3/3 pp. 325-349, 2005.
6. W. Heinzelman, A. Chandrakasan and H. Balakrishnam, "Energy-efficient communication protocol for wireless sensor networks" in the *Proceedings of the Hawaii International Conference System Sciences*, Hawaii, January, 2000.
7. A. Manjeshwar and D. P. Agrawal, "Teen: A Protocol foe Enhancement Efficiency in Wireless Sensor Networks" in the *Proceedings of the* 1^{st} *International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April, 2001.
8. S. Lindsey, C. S. Raghavendra, "PEGASIS: Power Efficient GAthering in Sensor Information Systems" in the *Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, March, 2002.
9. K. Akkaya and M. Younis, "An Energy-Aware QoS Routing protocol for Wireless Sensor Networks" in the *Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003)*, Providence, Rhode Island, May, 2003.
10. X.Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a library for Parallel Simulation of Large-scale Wireless Networks" in *Proceedings of the* 12^{th} *Workshop on Parallel and Distributed Simulations (PADS'98)*, May 26-29, 1998.
11. T. He et al., "SPEED: A stateless protocol for real-time communication in sensor networks" in the *Proceedings of International Conference on Distributed Computing Systems* , Providence, RI, May 2003.