A Priority-based Multi-path Routing Protocol for Sensor Networks

Yuzhe Liu[1, 2] Winston K.G. Seah[1]
[1]Networking Dept., Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119163
[2]Dept. of ECE, National University of Singapore, Singapore, 117576
{stuliuyz, winston}@i2r.a-star.edu.sg

ABSTRACT

A priority-based multi-path routing protocol (PRIMP) is proposed for sensor networks to offer extended network lifetime and robust network fault tolerance. Extensive simulations validate that PRIMP exhibits significantly better performance in energy conservation, load-balancing and data delivery than its comparable schemes. Moreover, PRIMP addresses the slow startup issue occurred in data-centric routing schemes.

1. INTRODUCTION

A sensor network is composed of a collection of un-tethered sensors that are densely deployed in a target area. Sensors are small in size, battery-powered, and sometimes embedded [1]. Each sensor has only limited sensing, processing and computational capabilities. Applications are fulfilled through the collaboration of such sensors.

However, key issues like stringent energy constraint and vulnerability of sensors to dynamic environmental conditions, still remain to be addressed. They create a demand for energy-efficient and robust protocol designs with specific consideration of the unique features of sensor networks, such as data-centric naming and addressing convention [2], high network density, and power limitation.

In this paper, we present a priority-based multi-path routing protocol (*PRIMP*) for sensor networks to address the above issues. *PRIMP* aims to offer extended network lifetime, and to provide robust network fault tolerance capability, as well as addresses the "*slow startup*" problem that may occur in data-centric routing schemes.

2. RELATED WORK AND MOTIVATIONS

An extensive survey on sensor networks can be found in [1]. Here, some key data-centric routing schemes are discussed. *Directed diffusion* [3] has unique features of declarative *interest* dissemination, reinforcement-based path selection and in-network processing, all of which contributes to energy-efficiency and robustness. However, the overhead is still large, and fault tolerance poor in the face of frequent temporary node failures. *SAR* [5] and the scheme proposed in [4] employ multi-path routing to provide reliable data delivery, but at the cost of foregoing energy-efficiency. *SPIN* [6] addresses the deficiency of classic *flooding* by negotiation and resource adaptation. But the network-wide *meta-data* dissemination that is employed is too energy-expensive in the case of new events that need data to be drawn from the network frequently.

To provide extended network lifetime, *PRIMP* achieves high energy-efficiency by minimizing overheads from all major data sources; to provide robust network fault tolerance, a probabilistic multi-path routing strategy is employed in a load-balanced manner.

PRIMP also addresses the *slow startup* issue occurred in data-centric routing [3]. For time-critical applications, short startup time can be critical. We define *startup* as the network status when every *sink* begins to collect data after an application is launched. Slow startup issue can be illustrated in *directed diffusion* with a simple two-sink-one-source example. Suppose two sinks, *A* and *B*, initiate identical interests. If the interest initiated by *A* arrives earlier at the *source*, exploratory data (from the sensors) will be sent back via all the established *gradients*. If all possible gradients leading to *B* have not been established, the exploratory data will not reach *B*. Shortly afterwards, when the source receives the interest initiated by *B*, it will not send exploratory data any more, because the reply to such interest type has already been provided. Thus, *B* cannot reinforce a path to draw data from the source, without first receiving exploratory data. This situation will last until the next round of exploratory data invocation at the source. Since exploratory data is infrequently dispatched due to its energy-consuming nature, long startup time is experienced by *B*.

In this study, we assume that the information of sources is absent, compelled by a broad spectrum of applications in such scenario. The existence of a localization system at each sensor is also assumed, as it enables each sensor to obtain its current position.

3. PROTOCOL DESIGN

PRIMP can be generalized into two stages: interest dissemination stage and priority-based path selection stage.

3.1 Interest dissemination stage

In this stage, interest dissemination is initiated by sinks periodically, and aims to constantly establish the up-to-date multiple data paths from sources to each sink. These paths will be used for routing at priority-based path selection stage.

The *virtual source* technique (discussed below) is invoked *reactively* at this stage. It is triggered to update sinks' knowledge of sources information, or to re-explore the data paths from sources to sinks when necessary. When it is invoked, interest will be propagated towards *virtual sources*. This effects that the interest be disseminated to all the nodes in the network.

After a source receives an interest, it will send back matched data events, piggybacking the geographic information about sources. When a sink receives this information, its subsequently initiated interests are directionally diffused towards the targeted sources. Thus, only the data paths from sources to sinks get maintained constantly to overcome the transmission unreliability. The *confirm flag* bit in such interests will be set to acknowledge the source.

Virtual source technique will be constantly invoked in interest dissemination until the piggybacked information is obtained by sinks; and sources will constantly send data containing the piggybacked information until they detect that the *confirm flag* bit in the received interest is set.

3.1.1 *Virtual source* technique

The topology of the target area can be encompassed within a rectangular region, and can be divided into four sub-areas based on a sink's position, as shown in Figure 1.

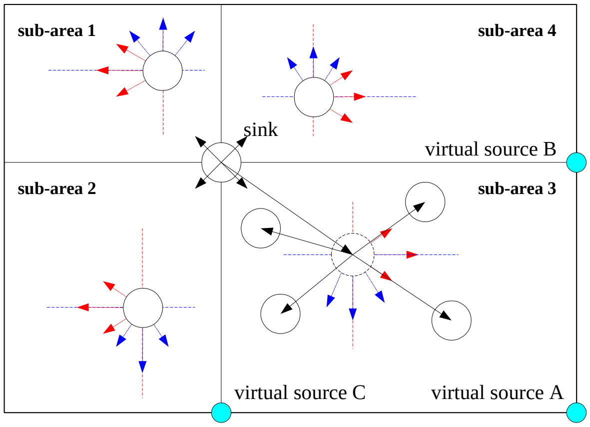


Figure 1. *Virtual source* technique

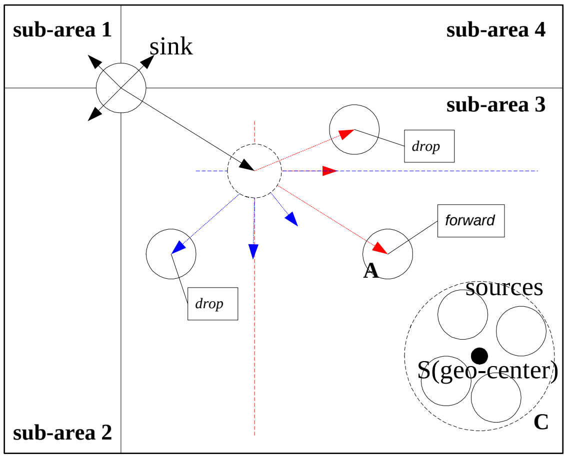


Figure 2. Interest dissemination toward sources

When the *virtual source* technique is invoked for interest dissemination, an interest generated at the sink is simply broadcast. After the interest arrives at one of the neighbors of this sink, virtual sources are selected for the packet. In *PRIMP*, virtual sources are always chosen to be at the corners of the current sub-area where this neighbor resides. Once virtual sources are determined, the interest packet will be disseminated towards them within the current sub-area. This effectively propagates the interests outward toward the four borders of the current sub-area, and forbids it from going inward towards the sink, as shown in Figure 1.

Once the location information of sources has reached the sinks, the *interest* dissemination strategy will take effect, and the *virtual source* technique will no longer be used. The subsequent interest dissemination will follow a different local rule: an interest receiver will forward the on-coming packet only if this node is approaching nearer to the sources in both latitude and longitude from its upstream neighbor, as illustrated in Figure 2.

3.1.2 Setting up gradient paths

When an interest is disseminated, a node may receive multiple copies of the interest from different neighbors due to high network density. Duplicate interest packets will not be forwarded, but will still be used to update the gradient cache. At each node, at most α "qualified" gradients are allowed to be set up and cached. The setup qualification of a gradient will be judged based on the *priority* information in the on-coming interest packet. Basically, the principle is to set up and cache the most energy-efficient α gradient paths leading to the sink at each hop of the data paths that are updated in every round of interest dissemination.

Thus, multiple ($\le \alpha$) gradients are set up and cached by nodes at each hop when an interest packet traverses through the network in a hop-by-hop fashion. Since at each hop, multiple downstream neighbors of a node may cache gradients towards this node, the established data paths from a source to a sink are in braided structure.

At each node, every cached gradient is tagged with two pieces of information: ***group id*** and ***priority***. *Group id* is generated together with an interest at a sink to specify that sink. The *group id* tag of a gradient indicates that its specified sink can be reached via this gradient. Identical interests from different sinks will be suppressed when they meet at intermediate nodes, but their group ids will be carried downstream by the forwarded interest of this type. Thus leads to significant energy conservation, as well as loop avoidance. *Priority* information can be either of type *accumulated hop count* or type *remaining power resource*. Unlike *group id*, *priority* is not constant. Every time a node is about to forward an interest, the *priority* information contained in the interest needs to be updated. By the time the interest arrives at a node and a gradient towards the forwarder is established, the *priority* information contained in the interest will be used to tag the established gradient.

3.1.3 Choosing and computing *priority* information

Gradients tagged with *accumulated hop count* are considered to be of *high priority*, while gradients tagged with *remaining power resource* are considered *low priority*.

As mentioned above, *priority* information contained in the received interest needs to be updated, i.e., its type needs to be determined and its value needs to be computed. To decide which type of *priority* information will be computed and enclosed in a to-be forwarded packet at a certain node *A*, as shown in Figure 3, both the energy level of *A* (marked as symbol "+" or "-") and the *priority* information of the cached gradients (marked as symbol *h* or *l*) shall be considered. The arrows in Figure 3 represent the directions of gradients. *UP_N* denotes the neighbor set pointed by the cached gradients at *A*. Symbols "+" and "-" denote the node's energy level is either above or below the self-configured energy threshold, symbolized "*good*" and "*poor*", respectively. Nodes may set same or different energy thresholds. Symbols *h* and *l* stand for high and low priority respectively, denoting the cached gradient towards an upstream neighbor ($\in$ *UP_N*).

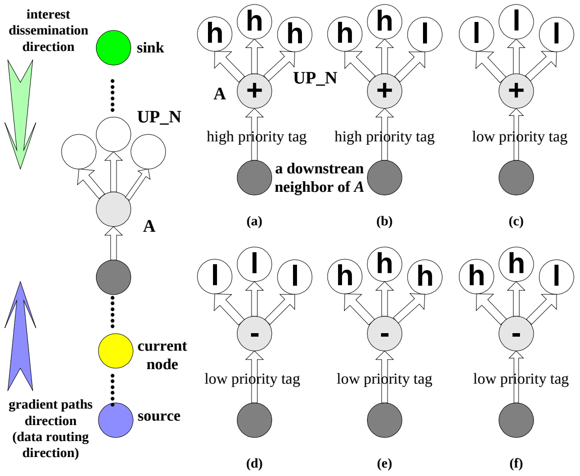


Figure 3. Choosing *priority* information

Figure 3 shows that a*ccumulated hop count* will only be used when "energy-sufficient paths" from the *current node* to the sink exist. In other words, for the paths from *current node* to the sink, at each hop (such as the hop from node *A* to *UP_N*), the energy level of the node is *good*, and gradients tagged with *accumulated hop count* are cached by the node, as demonstrated in Figure 2(a)(b). When no such energy-efficient paths exist, as in the cases shown in Figure 2(c)(d)(e)(f), *remaining power resource* will be chosen .

The value of *priority* information is computed in a cumulative approach. If a*ccumulated hop count* is chosen, at each hop, its value is computed as the average value of all the *accumulated hop counts* of the cached gradients, with the incremental of one. When *remaining power resource* is chosen at a certain hop: if some received interest copies contain *accumulated hop count*, those *high priority* accounts are converted to equivalent power values first (by multiplying count numbers by energy threshold); then, the result is obtained by incrementing the residual power of the current node to the average value of all the *remaining power resource* of the cached gradients and those converted values.

3.2 Priority-based path selection stage

As shown above, the updated *priority* tag value at a particular node indicates the energy resource condition of the data paths from this node to the sink. Our priority-based routing is based on this view. In this section, the principle of our path selection algorithm is demonstrated first with a single-sink-single-source scenario. Then, the extensions to multi-sink and multi-source scenarios are presented.

3.2.1 Basic routing principle

After an interest initiated by a sink arrives at a source, matched data events will be sent back. For each data event, multiple data paths will be used simultaneously to deliver it. Our data paths selection can be interpreted as gradient paths selection at each hop of routing. Gradient paths are selected based on their priorities. High priority gradients are preferred to low priority ones, if both kinds exist in the cache; low priority gradients can only be used when no high priority gradients are available.

When high priority gradients are used for routing, *PRIMP* prefers the gradients with smaller *accumulated hop count* tag values (shorter paths) for energy-efficiency. To achieve this, each cached gradient is assigned a weight based on its *accumulated hop count* value. Gradients selection is conducted probabilistically based on the weights of these gradients. The bigger the weight is, the more likely the corresponding gradient is used for routing. When low priority gradients are used for routing, similar weight assignment strategy based on *remaining power resource* value is employed. In this case, a gradient leading a path to the sink with more residual power will be favoured.

3.2.2 Routing in multi-sink scenarios

In a multi-sink scenario, when a data event arrives at a node, the node will route it along the cached gradients to all the sinks whose *group id*s exist in the matched interest entries in its cache. Since a gradient may be tagged with multiple *group id*s, paths to multiple sinks via a gradient may exist. Therefore, instead of selecting η different gradients for each sink and sending η copies of a data event along them, gradients selection must follow a simple rule—gradients that have been used η times will not be used any more. The gradients selection continues until every sink has been addressed η times. Thus, fewer copies of duplicate data events are transmitted, thus reducing overheads.

3.2.3 In-network processing in *PRIMP*

In *PRIMP*, for multi-source scenarios, data events will be suppressed if identical, and aggregated if supplementary to each other at the nodes inside or close to the phenomenon.

Data aggregation is critical in *PRIMP*. Since multiple data paths are simultaneously used to carry traffic, it is likely that multiple copies of a data event arrive at a same data-forwarding node. By dropping the later-arrived copies at this node, transmission overhead can be reduced.

4. PERFORMANCE EVALUATIONS

4.1 Performance metrics

Four metrics are evaluated in our study: *average dissipated energy* [8]; *average forwarded data* (measured as the average number of data events relayed by a data-forwarder for every distinct event delivered to sinks); *distinct-data delivery ratio*; and *application startup speed* (measured as the number of events seen by every sink after the launch of an application).

To study the influence of network densities, evaluations under various "high" densities are conducted. We randomly deploy 50–110 sensors in a fixed 150x150 m^2 target area, in increments of 20 nodes. Simulations are implemented in *ns-2* with 4 sinks and 4 sources with design parameters set as α =3 and η =2. Sources are located in a fixed source region of 50x50 m^2, while sinks are uniformly scattered across target area. Each sensor has a constant transmission range of 40 m. A 64-byte data event is sent every 0.5 s, 32-byte interest every 5 s, and 64-byte exploratory data event every 50 s. Comparisons are conducted among *PRIMP*, *directed diffusion* and the benchmark scheme — *flooding*.

A modified version of 1.6 Mbps *IEEE 802.11* MAC with realistic sensor network radio parameters [7] is implemented in our study, although its energy inefficiency makes is not entirely satisfactory. Its energy model is adopted such that power consumption for transmission, reception and idle state is 660 mW, 395 mW, and 35 mW, respectively. To minimize the variations on routing performance from MAC, no energy conservation strategy is introduced in this MAC protocol. By this, we tend to give the most conservative measurements on the advantages of *PRIMP* over other comparable schemes.

4.2 Simulation results

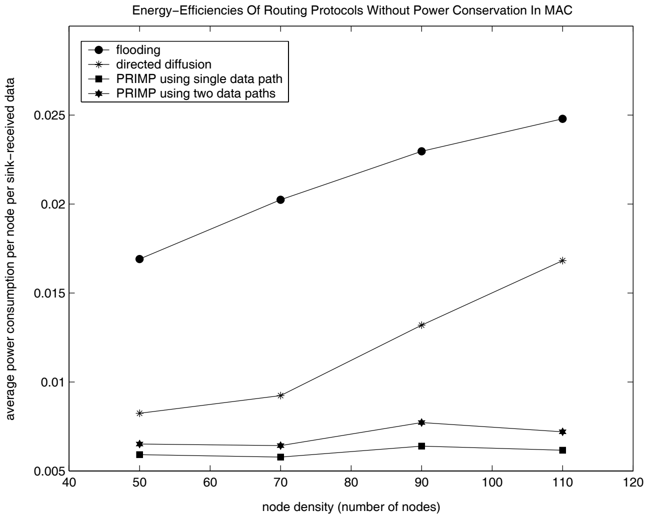


Figure 4. Energy-efficiency

Figure 4 shows the comparison of the three schemes on energy-efficiency. Unlike *directed diffusion* and *flooding*, the average dissipated energy of *PRIMP* is quite insensitive to the varying network densities. It is also observed that the energy-efficiency is still quite satisfactory when multi-path strategy is used, compared with the case when single-path strategy is adopted. As shown in Figure 4, even when multi-path strategy is used for routing, *PRIMP* still outperforms *directed diffusion* by 20% – 60%.

Figure 5 demonstrates the load balancing capability of three routing schemes. This metric reflects the long-term energy-efficiency and potential network fault tolerance. As shown in Figure 5, *PRIMP* (with single-path strategy) performs more than 2 times better in balancing the traffic load in most of density scenarios. When multi-path strategy is employed by *PRIMP*, the load balancing performance is not actually worsened as appeared in Figure 5. This is because every data-forwarding node just simply transmits more duplicate data events as compared to the case when single-path strategy is used for routing.

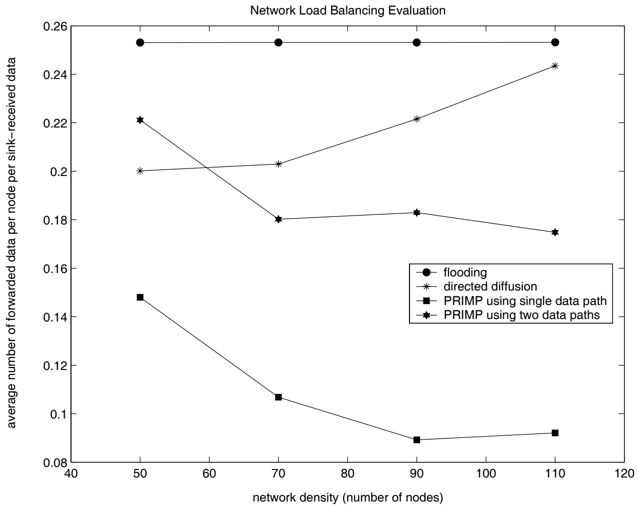


Figure 5. Load-balancing capability

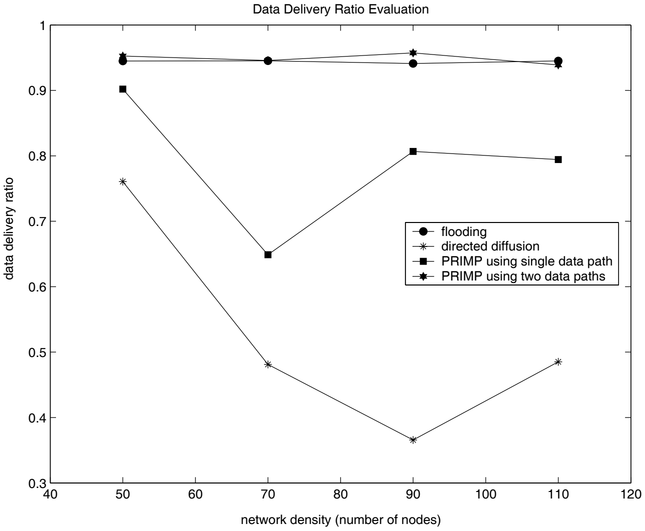
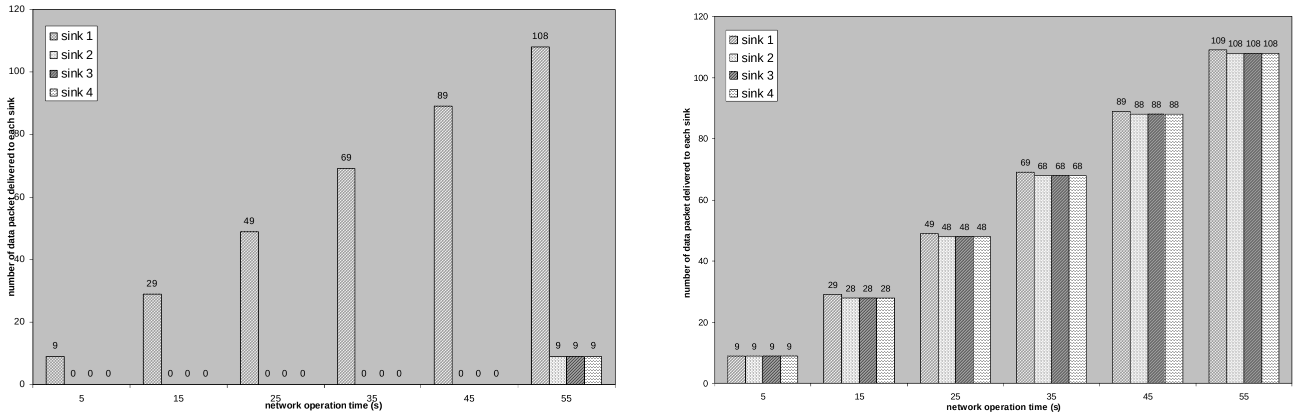


Figure 6. Fault tolerance capability evaluation

It is found in our study that without considering the influence from environmental conditions, the load balancing



(a) *Directed diffusion* **(b) *PRIMP***

Figure 7. Impact of *slow startup* problem

efforts of *directed diffusion*, if there are any, mainly come from MAC dynamics. This gives rise to an interesting observation: *CSMA*-based MAC protocols seem to show a less distinctive delay characteristic on different paths when network becomes denser. That is, with increasing network density, a *CSMA*-based MAC protocol tends to provide a more *isotropic* delay characteristic on the paths stretching out from a sink. As shown, when network density increases, the load balancing performance of *directed diffusion* becomes worse, implying that fewer shortest-delay data paths are used (delays on different data paths become less distinctive).

Figure 6 measures the fault tolerance capabilities of three schemes. Here, we introduce periodic, temporary (20 s) node failures to model the influence of dynamic environmental conditions on transmissions. Figure 6 shows that even with the single-path strategy, *PRIMP* still outperforms *directed diffusion* by at least 18%. The multi-path strategy significantly improves the fault tolerance and robustness of *PRIMP*, compared to the single-path strategy.

Figure 7 demonstrates the difference in data collections of different sinks. This measurement aims to shows the impact of slow startup problem on the data retrieval activity of different sinks. Suffering from the slow startup problem that is inherent in directed diffusion, sinks 2, 3 and 4 whose *interest* arrived later than that of sink 1, can only begin to collect data events long after the launch of the application. On the other hand, with *PRIMP*, all sinks are able to receive their desired data promptly.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented *PRIMP* which is designed to extend network lifetime and provide robust network fault tolerance. *PRIMP* is able to achieve this by (a) exploiting braided data paths from sources to sinks using an *on-demand virtual source* technique as a part of its interest dissemination strategy; (b) maintaining the paths through directional interest dissemination towards *sources*; and (c) probabilistic routing in a priority-based approach at each hop. Additionally, *PRIMP* addresses the slow startup problem occurred in data centric routing schemes. A key element of our ongoing work is focused the scalability issue. Moreover, it is noticed that the performance of *PRIMP* is a function of design parameters. Their impact on routing performance is also being studied.

REFERENCES:

[1]. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks: A Survey”, *Computer Networks*, Vol. 38, No.4, pp. 393–422, March 2002.

[2]. John Heidemann, Fabio Silva, and Chalermek Intanagonwiwat, “Building Efficient Wireless Sensor Networks with Low-Level Naming”, in *Proceedings of the Symposium on Operating Systems Principles*, Lake Louise, Banff, Canada, Oct. 2001, pp. 146–159.

[3]. Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Boston, MA, USA, Aug. 2000, pp. 56–67, ACM.

[4]. Xiaoyan Hong, Gerla, M., Hanbiao Wang, and Clare, L., “Load balanced, energy-aware communications for Mars sensor networks”, *Aerospace Conference proceedings*, 2002. IEEE, Volume: 3, 2002, Page(s): 3 - 1115 vol.3.

[5]. K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, “Protocols for self-organization of a wireless sensor network”, *IEEE Personal Communications*, October 2000, pp. 16–27.

[6]. W. R. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks”, *Proceedings of the ACM MobiCom 99*, Seattle, Washington, 1999, pp. 174–185.

[7]. W. J. Kaiser, “WINS NG 1.0 transceiver power dissipation specifications”, *Sensoria Corporation*, San Diego, CA.

[8]. A. Woo, and D. Culler, “A transmission control scheme for media access in sensor networks”, *Proceedings of ACM MobiCom 01*, Rome, Italy, July 2001, pp. 221–235.