Summary on ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies

Original paper by Alberto Cerpa and Deborah Estrin.

Sami Kauppinen
27.02.2005

1. Introduction

Wireless sensor network consists of a set of radio equipped sensor nodes that communicate with each other to perform distributed sensing tasks. Advances in microsensor and radio technology will enable these small nodes to be densely distributed. The large number of nodes deployed in these systems will preclude manual configuration, and the environmental dynamics will preclude design-time preconfiguration. In addition, the density of nodes causes redundancy since neighboring nodes may all sense practically the same condition and merely interfere with each others radio transmission, which decreases the systems overall performance. ASCENT is based on the idea that as the node density increases only a subset of the nodes are necessary to establish routing in the network. In ASCENT, each node assesses its connectivity and adapts its participation in the multihop network topology based on the measured operating region. The objective is to make use of the redundancy over time to extend the systems life.

The paper rules out the possibility that the adaptive configuration could be done from a central node. This is argued on the grounds that a single node cannot directly sense the conditions of nodes distributed elsewhere in space and consequently, other nodes would need to communicate detailed information about the state of their connectivity in order for the central node to determine who should join the multihop network. Since energy is a constraint and the environment is dynamic, distributed approaches are presented in the paper as an attractive and possibly the only practical approach because they avoid transmitting dynamic state information repeatedly across the network.

2. Distributed Sensor Network Scenario

The paper argues that wireless sensor networks must be designed to operate under the following conditions and constraints:

- Ad hoc deployment: Nodes cannot be expected to be deployed in a regular fashion. More importantly, uniform deployment does not correspond to uniform connectivity because of unpredictable propagation effects when the node antenna is close to the ground and other surfaces.
- Energy constraints: At least some significant subset of the nodes will be untethered for power as well as communications and therefore the system must be designed to expend as little energy as is possible in order to maximize the networks lifetime.
- Unattended operation under dynamics: The anticipated number of elements in the network will preclude manual configuration, and the environmental dynamics will preclude design-time preconfiguration.

In addition, the paper argues that in the case of wireless sensor networks, it will be far easier to deploy larger number of nodes initially than to deploy additional nodes or additional energy reserves at a later date. The ASCENT method exploits the resulting redundancy in order to extend system lifetime.

The proper selection of nodes that will operate actively on the network is important. If too few of the deployed nodes are used, the distance between neighboring nodes will be too long and the packet loss rate will increase or the energy required to transmit the data over the longer distances will be prohibitive. If all deployed nodes are used simultaneously, the system will be expending unnecessary energy at the best and, at worse, the nodes may interfere with one another by congesting the channel.

The ASCENT protocol reacts when high packet loss is detected at the links. However, it does not detect or repair network partitions of the underlying topology. Partitions are more prevalent when node density is low, in which case the ASCENT protocol is not applicable because all the nodes will be needed to form an effective network. Partitions can occur even in dense networks when a group of nodes is not functioning for some reason. The paper points out that when network partitions do occur, complementary methods are needed. These methods are left to future work.

3. ASCENT Design

ASCENT adaptively elects “active” nodes from all nodes in the network. Active nodes stay awake all the time and perform multihop packet routing, while the rest of the nodes remain “passive” and periodically check if they should become active.

Situation where the receiver is at the limit of radio range and experiences very high packet loss is called a *communication hole*. In that case the receiver starts sending *help messages* to signal neighbors that are in listen-only mode – also called *passive neighbors* – to join the network. When a neighbor receives a help message, it may decide to join the network. When a node joins the network, it signals the existence of a new *active neighbor* to other passive neighbors by sending a *neighbor announcement message*.

3.1. ASCENT State Transitions

In ASCENT, nodes are in one of four states: *sleep*, *passive*, *test*, and *active*. State transition diagram is shown on fig. 1.

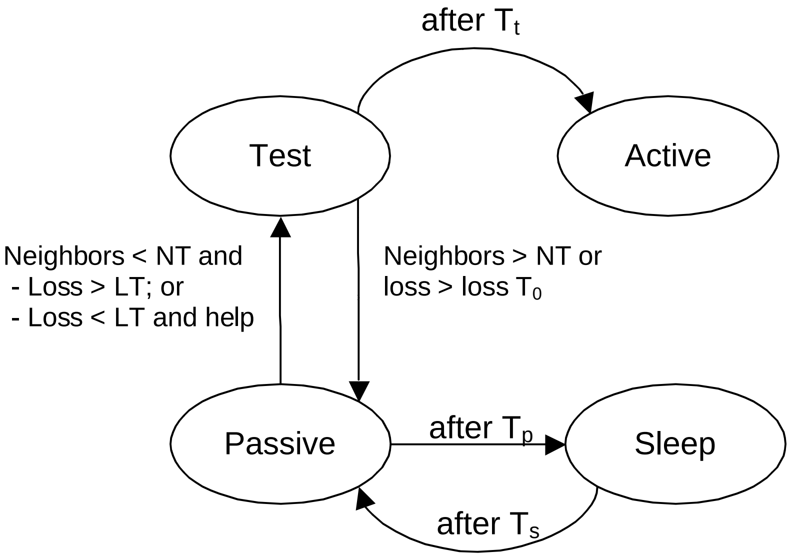

Figure 1. ASCENT state transitions

Initially, a random timer turns on the nodes to avoid synchronization. When a node starts, it initializes in the *test state*. Nodes in the *test state* exchange data and routing control messages. In addition, when a node enters the *test state*, it sets up a timer T_t and sends *neighbor announcement messages*. When T_t expires, the node enters the *active state*. If, before T_t expires, the number of active neighbors is above the *neighbor threshold* (NT) or if the average *data loss rate* (DL) is higher than the average loss before entering in the *test state*, then the node moves into the *passive state*. If multiple nodes make a transition to the *test state*, then we use the node ID in the announcement message as a tie breaking mechanism (higher IDs win). The intuition behind the *test state* is to probe the network to see if the addition of a new node may actually improve connectivity.

When a node enters the *passive state*, it sets up a timer T_p and sends *new passive node announcement messages*. This information is used by active nodes to make an estimate of the total density of nodes in the neighborhood. Active nodes transmit this density estimate to any new passive node in the neighborhood. When T_p expires, the node enters the *sleep state*. If, before T_p expires, the number of neighbors is below NT and either the DL is higher than the *loss threshold* (LT) or DL is below the *loss threshold* but the node received a *help message* from an active neighbor, it makes a transition to the *test state*. While in *passive state*, nodes have their radio on and are able to overhear *all* packets transmitted by their active neighbors. No routing or data packets are forwarded in this state since this is a listen-only state. The intuition behind the *passive state* is to gather information regarding the state of the network without causing interference with the other nodes. Nodes in the *passive* and *test states* continuously update the number of active neighbors and data loss rate values. Energy is still consumed in the *passive state* since the radio is still on when not receiving packets. A node that enters the *sleep state* turns the radio off, sets a timer T_s, and goes to sleep. When T_s expires, the node moves into *passive state*. Finally, a node in *active state* continues forwarding data and routing packets until it runs out of energy. If the *data loss rate* is greater than LT, the active node sends *help messages*.

3.2. ASCENT Parameters Tuning

ASCENT has some parameters that affect its final behavior. The selected parameter values effect the energy consumption and on the other hand the reaction time in case of dynamics.

The *neighbor threshold* (NT) value determines the average degree of connectivity on the network. In the study presented at the paper, the NT value was set to 4. The loss threshold (LT) determines the maximum amount of data loss an application can tolerate before is requests help. The value of LT is very application dependent. In the study, the value of LT was set to 20 percent.

The test timer T_t and the passive timer T_p determine the maximum time a node remains in the test and passive states, respectively. In the study, the T_p value was set to 2 minutes and T_t to 4 minutes. Similarly, the sleep timer Ts represents the amount of time the node sleeps to preserve energy. ASCENT uses an adaptive probabilistic mechanism in order to determine the optimal relationship between the T_p and T_s timers.

3.3. Neighbor and Data Loss Determination

The number of active neighbors and the average data loss rate are values measured locally by each node while in passive and test state. The paper defines a neighbor as a node from which a certain percentage of packets are received over time. A history window function (CW) is used to keep track of the packets received from each individual node over a certain period. In addition, fixed or dynamic *neighbor loss threshold* (NLS) is used. An active neighbor is a node with link packet loss smaller than the NLS. The value of NLS was calculated with formula `NLS = 1 – 1/N`, where N is the number of active neighbors.

Each node adds unitary monotonically increasing sequence number to each data and control packet transmitted. This permits neighbor link loss detection when a sequence number is skipped. The paper assumes that in addition, application data packets also have some mechanism to detect losses.

The average *data loss rate* (DL) is calculated based on the application data packets. Data losses are detected using sequence numbers. Data loss is considered if the packet was not received from any neighbor during a certain configurable period of time.

3.4. *ASCENT Interactions with Routing*

ASCENT runs above the link and MAC layer and below the routing layer. All ASCENT control messages are broadcast locally to the neighbors and they do not require any multihop forwarding scheme. ASCENT is not a routing or data forwarding protocol. ASCENT simply decides which nodes should join the routing infrastructure.

ASCENT nodes become active or passive independent of the routing protocol running on the node. Currently, if a node becomes passive, ASCENT depends on the routing protocol to quickly reroute the traffic. This may cause some packet loss and therefore an improvement that has not been implemented is to inform the routing protocol of ASCENT's state changes.

The paper emphasizes that, even though the ASCENT algorithm is discussed in some detail, much experimentation and evaluation of the various mechanisms and design choices is necessary before the robustness, scale, and performance of self-configuration is fully understood.

4. Performance Evaluation

ASCENT performance is evaluated using the following metrics:

- *One-Hop Delivery Rate* measures the percentage of packets received by any node in the network, and it indicates the effective one-hop bandwidth available to the nodes. When all the nodes are turned on – the *Active case* – packet reception includes all nodes. In the ASCENT case, it includes all but the nodes in the *sleep state*.
- *End-to-End Delivery Rate* is the ratio of the number of distinct packets received by the destination to the number originally sent by the source, and it provides an idea of the quality of the paths in the network and the effective multihop bandwidth.
- *Energy Savings* is the ratio of energy consumed by the *Active case* to the energy consumed by the ASCENT case.
- *Average Per-Hop Latency* measures the average delay in packet forwarding in a multihop network, and it provides an estimate of the end-to-end delay in packet forwarding.

Both simulations and experiments with actual hardware components were used to evaluate the performance of ASCENT. The experiments were run with different densities ranging from 5 to 40 nodes. In the study, density was defined topologically, i.e. the density of nodes is defined by the average degree of connectivity of all the nodes in the experiment and not by their physical location. The different levels of density were achieved by adjusting the transmission power. The average number of hops in the topologies obtained by the method was three. The simulations replicate the same scenario, but using densities ranging from 5 to 80 nodes. The average number of hops in the simulations was six. The results were obtained by taking the average of three experimental trials, and for simulations by average of five simulation trials.

Flooding was used as the routing protocol. This was because the authors lacked other routing protocol implementations, and it was also argued that several routing algorithms still use some form of flooding as part of the routing strategy. When receiving a packet the flood routing module wait for a random time between zero and the maximum randomization interval before forwarding the packet.

Some of the results are obtained using version of the ASCENT that uses *adaptive state timers*. However, *adaptive state timer* is not clearly defined in the paper. It probably refers to the adaptive probabilistic mechanism that is used to determine the optimal relationship between the T_p and T_s timers. The paper implies that the T_p timer has a fixed value and only the T_s timer is adaptive.

4.1. Network capacity

The paper reports that with one-hop delivery rate, no important differences between the expected analytical (not presented in this summary) and simulated performance and the performance using real radios up to densities of 40 nodes. When the density increases ASCENT performs better than the *Active case*. Similar results are reported with the end-to-end delivery rate.

It is worth mentioning, that these results reflect the worst-case scenario of flooding contention with increased density. The paper reports no results for cases when non-flooding routing strategy is used, but leaves it to future work.

4.2. Energy Savings

Results are presented using two versions of the ASCENT algorithm, one with fixed and the other with adaptive state timers. It is reported that when using ASCENT with fixed state timers, the energy savings do not increase proportionally as the density increases. The opposite could be expected, since the number of active nodes remains constant as the density increases. The explanation is that the energy consumption with increased density is dominated by the passive-sleep cycle of the passive nodes and not energy consumed by the active nodes. ASCENT provides a factor of 4 in energy savings in this case. When the adaptive state timers were used, the energy savings are reported to increase as the density increases.

4.3. Latency

ASCENT increases the average per-hop latency when compared to the active case. When the density increases, the active case reduces the average per-hop latency because there is a larger probability of a node picking a smaller random interval to forward the packet. In the case of ASCENT the number of nodes able to forward the packets remains constant and consequently the average per-hop latency tends to remain stable as the density increases.

5. Conclusions and future work

The paper describes the design, implementation, analysis, simulation and experimental evaluation of ASCENT. The paper concludes by stating that ASCENT has the potential for significant reduction of packet loss and increase in energy efficiency. Also ASCENT mechanisms are concluded responsive and stable under systematically varied conditions.

The authors mention interaction of ASCENT with new MAC mechanisms to be one subject of their future work. In addition, load balancing techniques to distribute the energy load and the use of different routing strategies with ASCENT is also left for future work.

6. Critique

The paper describes the ASCENT mechanism in some detail, but some important details are left vague. First, the control messages and state transitions could have been described more precise. Second, the data loss determination based on the packet sequence number is also left somewhat unclear.

It seems that the ASCENT method works fine with selecting only a subset of nodes to be used in the routing infrastructure. For routing purposes this clearly is a good solution. But it comes to mind, that the very reason that these nodes are deployed is to sense events in the surrounding environment and to receive data from them. Using just a subset of the deployed nodes may not always be what is desired even though the nodes are deployed densely. The paper motivates the use of a subset of nodes by stating that it is easier to deploy large number of nodes densely initially than to deploy additional nodes at later date. But instead of deploying nodes densely and using just a subset of them, wouldn’t it be possible to deploy less nodes and putting more batteries to them? For example, instead of deploying 10 nodes closely to each other and using just one of them, it might be possible to deploy just one node with 10 batteries.

References

[1] Alberto Cerpa, Deborah Estrin: “ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies”. IEEE transactions on mobile computing 3 (2004). 272-285.