Efficient and Practical Query Scoping in Sensor Networks

Henri Dubois-Ferrière*
EPFL
Lausanne, Switzerland
henri.dubois-ferriere@epfl.ch

Deborah Estrin
UCLA/CENS
Los Angeles, CA 90095
destrin@cs.ucla.edu

Thanos Stathopoulos
UCLA/CENS
Los Angeles, CA 90095
thanos@cs.ucla.edu

Abstract

In this work we propose Voronoi scoping, a distributed algorithm to constrain the dissemination of messages from different sinks. It has the property that a query originated by a given sink will be forwarded only to the nodes for which that sink is the closest (under the chosen metric). Thus each query is forwarded to the smallest possible number of nodes, and per-node dissemination overhead does not grow with network size or with number of sinks. The algorithm has a simple distributed implementation and requires only a few bytes of state at each node. Experiments over a network of 54 motes confirm the algorithm's effectiveness.

1 Introduction

Environmental monitoring is a driving application for sensor networks. Such applications often use a *data gathering tree*, rooted at the basestation (or *sink*). However, a single-sink system scales poorly with network size: data packets must traverse an increasing number of hops as the network diameter grows (thus reducing delivery ratio and increasing energy consumption), and nodes near the root of the tree carry a disproportionate amount of traffic. Thus we expect that many large-scale data-gathering networks will employ a tiered architecture with *multiple sinks*, placed at different points in the coverage area. For example, the deployment plan [1] driving this work is a habitat monitoring sensor network with approximately 100 Berkeley motes and 5 Stargate [2] microservers which connect to the internet via a 802.11 transit network.

This paper addresses the problem of efficient *query dissemination* and *tree construction* in a multi-sink data-gathering network. The two tasks are respectively distributing a query to sensor nodes in the system, and constructing paths from each source to a nearby sink.

We propose a simple and practical algorithm called *Voronoi scoping*. The algorithm disseminates messages from each sink only to the subset of nodes that lie within the sink's Voronoi cluster, that is, the set of nodes for which that sink is the closest. The algorithm uses only local state and does not require nodes to have any knowledge of network topology, nor of number and location of sinks.

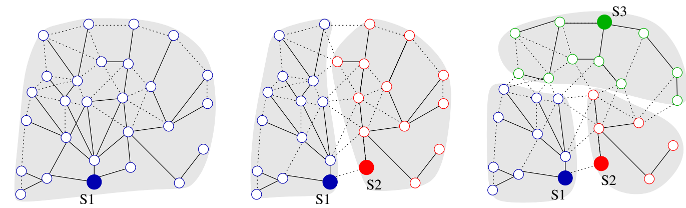


Figure 1: Voronoi clusters for $1, 2, 3$ sinks. Shaded regions represent the cluster associated with each sink under a hop count distance metric. The underlying topology is shown with dotted lines.

2 Algorithm

Each node keeps a record of its closest sink and of the network distance to that sink. When a message arrives from a sink, the recipient checks if the distance traversed by the packet is less than the current estimate of closest sink distance. If so, the node updates its closest sink and parent entries and resends the message. The algorithm pseudocode is given in Fig. 2. The algorithm and its properties are discussed in more detail in [3], as well as a generalisation where each node considers the distance to n closest sinks.

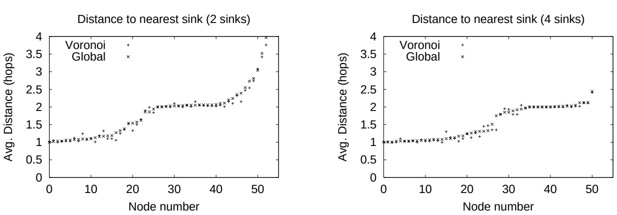
3 Experiment Setup

We ran experiments over a network of 54 MICA1 motes with 1, 2, 3, and 4 sinks. For each setup we ran identical experiments with Voronoi scoping and with global scoping. We also ran a second, identical set of experiments with

*The work presented in this paper was done while the author was visiting UCLA. It was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

Algorithm *Voronoi*(k, d, n)
(∗ Called each time a query packet is received. ∗)
Input: A sink k, a distance d, a neighbor n
(∗ k and d are read off incoming packet header. ∗)
(∗ n is the neighbor which transmitted the packet. ∗)
(∗ d_min is the distance to closest known sink. ∗)
$\triangle \leftarrow d - d\_min$
if $(\triangle \leq 0)$
  (∗ Update if closer or equal dist. than to current sink. ∗)
  then
    $parent \leftarrow n$
    $closest\_sink \leftarrow k$
    $d\_min \leftarrow d$
if $\triangle < L$ or $(\triangle = 0$ and $k = closest\_sink)$
  (∗ Reforward if in Voronoi cluster or overlap region. ∗)
  then
    $increment\_distance\_on\_packet()$
    $reforward\_packet()$

Figure 2: Voronoi scoping algorithm. Initially d_min is set to ∞. L is a fixed algorithm parameter. When $L = 0$, we have disjoint scopes; this is the setting we implicitly refer to as 'Voronoi scoping'.



(a) Network with 2 sinks.

(b) Network with 4 sinks.

Figure 3: Average distance (in hops) to closest sink, as a function of node id. The nodes are sorted by increasing distance to their closest sink under global scoping. Both algorithms result in near-equivalent distances for each sink, indicating that the source-sink route lengths are not modified by introducing Voronoi scoping.

link filtering. There were therefore four protocol combinations for each sink configuration, giving a total of 16 different configurations. For each of these 16 configurations we ran 5 identical experiments, each lasting 30 minutes. This amounts to approximately 40 hours of run time for the results presented here. During an experiment sinks periodically flooded query packets, and sources injected data packets. We implemented Voronoi scoping within one-phase pull directed diffusion [4], running under EmStar [5].

For completeness we give all protocol constants. Each 30 minute run included a 5 minute warmup time during which statistics were not recorded. Sinks flooded a query at a fixed interval of 120 seconds (with a phase shift between sinks to avoid synchronisation). Each source generated data packets at random, exponentially distributed intervals; the mean was selected so as to have 1 data packet per second generated over the whole network.

3.1 Connectivity Dynamics and Link Filtering

Given vagaries of RF propagation such as interference, path loss, and fading, wireless connectivity is often time-varying and/or lossy, especially with simple, low-power devices such as sensor nodes. Measurement-based studies (e.g., [6]) have shown that real networks behave differently from a unit-disk graph in many aspects. Given these observations, we should emphasize that it is not a priori obvious that our scoping algorithm will behave as predicted when subjected to a real environment. As an example of how real RF conditions can affect the simplest network primitive [7] shows how trees obtained through a simple reverse-path flooding approach can have unexpected topologies.

In wireless networks, selecting a parent by minimizing hop count only can result in selecting long-range, high-loss links, whereas a longer route with shorter-range, more reliable hops could be preferable. To address this problem, we run a neighborhood tracking module at each node which estimates the quality of links to peers. The obtained values are then used to perform link filtering: links with quality below a fixed threshold are not considered in the routing process.

3.2 Results

Dissemination Overhead. The first quantity of interest was the overhead required to disseminate the periodic query messages from sinks. This overhead is represented in Fig. 4, comparing Voronoi scoping with global scoping both with and without link filtering. As expected, Voronoi scoping has same overhead as global scoping when a single sink is used. Then, with each additional sink, overhead increases linearly for global scoping; on the other hand it remains roughly constant for Voronoi scoping. Even at the moderate size of our network, this amounts to a substantial difference: with 4 sinks, Voronoi scoping reduces overhead by a factor of 3 when using link filtering, and 6.5 without.

Another observation is that for both global and Voronoi scoping, link filtering reduces packet transmissions by a factor of 2 to 3. This is a nice side-effect since the primary objective of link filtering is to improve the quality of routes obtained.

Topology characteristics. We now examine some topological aspects of the resulting data-gathering trees in order to see if Voronoi scoping introduces any distortion compared to the trees obtained with global flooding. Note that the data-gathering trees are not static because nodes may change parent during the experiment.

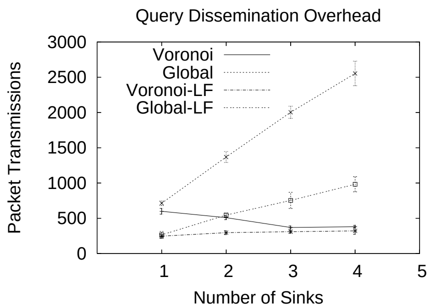


Figure 4: Comparison of query dissemination overhead for Voronoi scoping and global flooding, as a function of number of sinks.

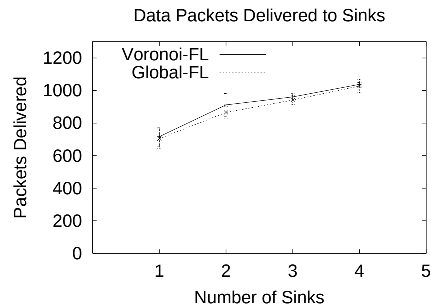


Figure 5: Comparison of data delivery for Voronoi scoping and global flooding, as a function of number of sinks. Delivery is measured as the total number of data packets successfully received at any sink.

Since the tree topology often varies over the course of an experiment, we cannot directly compare the trees obtained in different runs. One way to characterize these time-varying trees is to compute, for each node, the average distance (in hops) to the closest sink. We computed the time-weighted average distance for each node. We then sorted the nodes by average distance (under global flooding) and plotted the results for both protocols in Fig. 3.

These plots indicate that nodes have paths of similar lengths with Voronoi and global dissemination. Indeed, for 85% of nodes, the difference in average distance is less than 0.15; for the remaining nodes the difference is less than 0.25 (barring one outlier at 0.4).

We also observe in Fig. 3 a plateau-like levelling at hop-counts 1 and 2, indicating that most nodes' route lengths remain quite stable. Some nodes have an average distances which is "far" from an integer value (for example 1.5). Such nodes therefore were less stable and alternated between closest sinks at different distances. The plots indicate that stability is not adversely affected by Voronoi scoping, since nodes are as close to an integer-valued average distance as with global flooding.

Application performance. We examined *data delivery* as an indication of application performance. Counting the number of data packets succesfully delivered to sinks allows us to see, for example, if node starvation (Sect. 3.1) is occuring more often with Voronoi scoping. Data delivery is plotted in Fig. 5, for experiments both with and without link filtering. In both cases, we see that Voronoi scoping does not impact application performance. In fact the data delivery ratio appears slightly higher with Voronoi scoping; though this cannot be considered as definite evidence given the confidence intervals obtained.

4 Conclusions

We have presented Voronoi scoping, a technique to control dissemination of packets originated from multiple sinks in a network. Flooding overhead remains constant independantly of the network size and number of sinks. This is particularly valuable in the context of sensor networks, where energy is the limiting resource. We implemented and tested Voronoi scoping over a network of 54 motes; results confirm good efficiency compared to global flooding whilst maintaining a similar level of application performance.

References

[1] CENS/UCLA: Extensible sensing system: An advanced network design for microclimate sensing. http://www.cens.ucla.edu/Project-Descriptions/Extensible%20Sensing%20System/index.html (2003)

[2] Crossbow: X-scale single board computer and wireless networking platform. http://www.xbow.com/Products/XScale.htm (2003)

[3] Dubois-Ferriere, H., Estrin, D.: Efficient and practical query scoping in sensor networks. In: CENS Technical Report 0039. (2003)

[4] Heidemann, J., Silva, F., Estrin, D.: Matching data dissemination algorithms to application requirements. In: Proceedings of ACM Sensys, Los Angeles, USA (2003)

[5] Girod, L., Elson, J., Cerpa, A., Stathopoulos, T., Ramanathan, N., Estrin, D.: Emstar: a software environment for developing and deploying wireless sensor networks. In: Proceeding of the 2004 USENIX Technical Conference. (2004)

[6] Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proceedings of ACM Sensys, Los Angeles, USA (2003)

[7] Ganesan, D., Krishnamachari, B., Woo, A., Culler, D., Estrin, D., Wicker, S.: Complex behavior at scale: An experimental study of low-power wireless sensor networks. In: UCLA Computer Science Technical Report UCLA/CSD-TR 02-0013. (2003)