Location-aware In-Network Monitoring in Wireless Sensor Networks

Volker Turau and Christoph Weyer
Department of Telematics, Technische Universität Hamburg-Harburg
Schwarzenbergstraße 95, 21073 Hamburg, Germany
{turau,c.weyer}@tuhh.de

Abstract: This paper presents a data-centric and location-aware procedure to perform in-network evaluation of queries in sensor networks. The algorithm is a hybrid of geographic hash tables and in-network aggregation. It increases data availability and accounts for topology changes and generates mostly local network traffic.

1 Introduction

Wireless Sensor Networks offer new prospects for pervasive monitoring of natural and industrial environments. The stringent resource constraints of sensor nodes rule out many solutions known from classical distributed systems. The utility of sensor networks stems primarily from the data it gathers. This data can be retrieved continuously, in an ad hoc style or the data delivery to an external node can be initiated by an event triggered by a sensor node. In most cases the requested data will be an aggregation of values from different sources. Producing such digests recognizing the energy constraints poses unique challenges. Node failures and moving nodes introduce additional complexity. Recently, much attention has been focused on fault tolerant in-network aggregation of time series data to produce data statistics minimizing energy consumption. In this paper a monitoring infrastructure for sensor networks providing a data-centric view based on a hybrid data-dissemination technique is proposed. The algorithm uses geographic hash tables for in-network aggregation and accounts for node failures and topology changes.

2 Data-Dissemination

At least three distinct data-dissemination and aggregation procedures have been proposed in the literature. A straightforward implementation is to route all sensor readings to an external station hosting a database. While query evaluation can resort to the infrastructure provided by the database, the approach results in a high communication effort. The high communication rate is solely tolerable in cases where only indispensable data is delivered to an external node and this data is used in extensive analyses. To retain an SQL-style query language the TAG project uses a single virtual table, the actual aggregation is performed inside the network in two stages [MFHH02]. In the first stage queries are pushed down into the network and a routing tree is set up. In the second stage partial aggregates are computed and routed up from children to parents. This technique relies on some variant of global flooding such as *Directed Diffusion* [IGE00]. Besides the high-energy consumption due to the many broadcasts the approach also suffers from node failures. In case a single node fails, the entire network below that node is excluded from the aggregation leading to incorrect results. A set of aggregation operators, suitable for this kind of processing is identified in [MFHH02] and includes those known from SQL. Apart from the described variants *direct delivery* and *partial aggregation* there is a third option: *packet merging*. In case many nodes send packets simultaneously to the same target, data items of nodes encountered en route can be placed inside the payload of the current package (if payload size permits). This reduces communication cost since the packet overhead is only paid once for a group of data items.

The purpose of sensor networks is to provide data describing the complete or partial state of a geographic region. Hence, the identity of an individual node is of minor importance. This requires to move from the point-to-point communication abstraction of the internet protocols to a data-centric abstraction allowing the formulation of requests for data independently of node identifiers using for example geographic constraints. In the *data-centric storage* model data items are *named* and communication abstractions refer to these names rather to node addresses. Requests for data-items are routed to appropriate nodes, where the relevant data can be found. This way logical access points are defined. Geographic hash tables (GHT) as defined in [RKS+03] describe a solution for this type of storage, names of data items are hashed to geographic locations inside the area covered by the network and the data is stored at the node closest to that location. GHT is built on top of GPSR [KK00], a geographic routing system for multi-hop wireless networks. GHT is often used as a base layer to support higher semantic concepts. DIFS is an extension of GHT to efficiently support range queries, that is to say queries where only events with attributes in a certain range are desired [GEG+03] and R-DCS is another data-centric storage concept providing a higher degree of scalability and resilience [GGC03].

This paper extends the data-centric storage concept of GHT to perform in-network evaluation of queries affecting a high number of sensor nodes for monitoring and alarming purposes (as opposed to ad hoc queries). The presented procedure is a hybrid of GHT and in-network aggregation with improved scalability. It generates very local network traffic, increases data availability and accounts for topology changes caused by node failures, deployment of new nodes, decreasing communication radii, and moving nodes.

3 Algorithm

The proposed monitoring system supports two main operations: (1) alarming (i.e. the network generates an event if a specified expression of sensor readings evaluates to true) and (2) repeated and long running queries. Ad hoc queries are not considered, the overhead to set up the infrastructure is too high. The algorithm is applicable to queries including an explicit definition of a geographic target region. Only networks satisfying the following preconditions are considered: (1) individual nodes know their geographic location and (2) the nodes relevant for a query are located in a region that is significantly smaller than the region covered by network.

Like other distributed hash tables GHT supports two operations: *put(location, value)* and *get(location)*[1]. The value is stored at the node L closest to the specified location (called *home node*) and at all replicas in the vicinity of L (*home perimeter*), in most cases there will be two replicas. To support query evaluation GHT can be used as follows: Consider the smallest area A_Q encompassing all sensor nodes relevant to the query Q. Let L_Q be the center of A_Q (using any reasonable measure, such as minimal average distance to all nodes). The home node for L_Q uses restricted flooding to distribute the query to the nodes in the target area A_Q. All participating sensor nodes periodically make a call *put*(L_Q, *value)* to store the value in a suitable data structure hosted by the home node (and in the replicas). Sensor nodes remain passive after the timer has expired in case the measured sensor value has not changed compared to the last measurement. Having access to all data relevant to Q, the home node can evaluate the query and if required initiate an action (home nodes contain the code to process queries). External nodes have direct access to the result of the query using the operation *get*.

The described procedure can be optimized by changing the implementation of the *put* operation to perform packet merging while traversing the network (see figure 1). Each node has a timer, if the value of this timer has reached 0 the *put* operation is invoked and the timer is reset. Nodes visited en route include their data in the current packet and also reset their timers. This way, the number of *put* calls will be minimal after an initial phase of adjustment and the system will automatically reach a stable state after changes of the network topology.

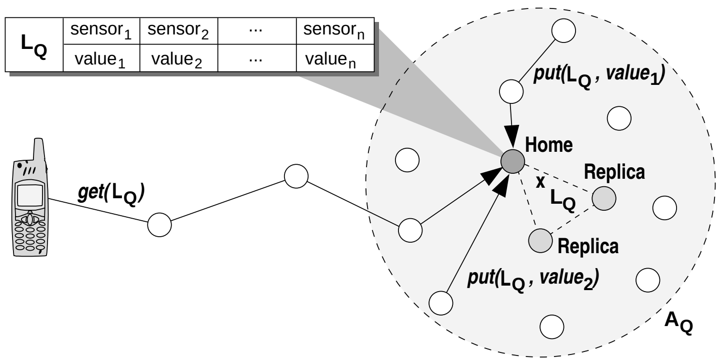

Figure 1: GHT in combination with packet merging

This approach has two shortcomings: (1) the logical storage point may become a bottleneck (hot spot) in the network leading to a non-uniform energy utilization and (2) if

[1]The original GHT operations use keys instead of locations. A hash function maps a key to a location in the area covered by the network. The usage of GHT in this work does not rely on this feature.

all nodes of the home perimeter fail simultaneously (cluster failure), the procedure fails totally and the query must be newly instantiated. To remedy these disadvantages a hierarchical system is proposed. The region encompassing the nodes involved in the query is partitioned into connected regions $R_1, \ldots, R_r$, each comprising roughly the same number of nodes. The geographic center L_i of each region R_i is used as a logical storage point for the sensor readings of the sensors of that region. The nodes of a region follow the procedure sketched above to report their sensor values to the defined center. In addition a location L_Q for the query itself is selected. The home nodes of the centers of the regions either partially evaluate the query and report the result to the global query location or if partial evaluation is not possible, all collected values are reported to the global query location. Both cases make use of the *put* operation (see figure 2). The introduction of the intermediate storage points reduces the number of *put* operations with target L_Q and thus decreases the formation of hot spots and leads to a more uniform energy consumption. Even so the total number of calls of the operation *put* increases, a reduction of the total number of communication hops can be expected. If the number of nodes increases the described algorithm can be extended by introducing intermediate layers of logical storage points between the storage points of the regions and the global query storage point. This way a hierarchy of logical storage layers is built. To compensate for cluster failures, the global query location can be replicated in a zone, different from the zone that contains location L_Q, similar to R-DCS [GGC03].

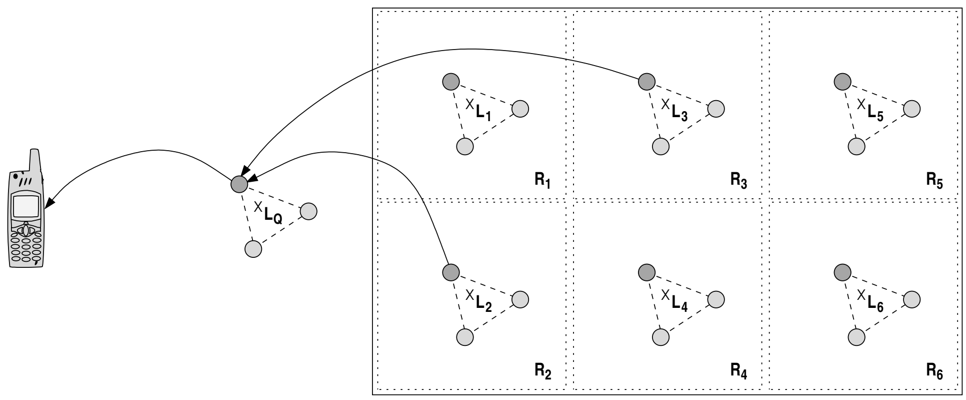

Figure 2: Hierarchical GHT in combination with partial evaluation

Due to space limitations it is only possible to give a sketch of the processing. To control the liveliness of nodes the timers t_{TTL} and $t_{refresh}$ are used:

1. When a node n in region R_i wakes up then
    (a) n performs measurements required by query Q and stores the value in v
    (b) if v has changed since the last measurement or if v was constant for a period longer than t_{TTL} n calls $put(L_i, v)$ and t_{TTL} is reset

2. When the home node h_i of region R_i wakes up then
   (a) h_i removes values older than t_{TTL} and computes the query result r_Q
   (b) h_i calls $put(L, r_Q)$, where L is the parent location
   (c) if the number of alive nodes has fallen below a threshold or the time $t_{refresh}$ has passed since the last update, h_i sends Q again to all nodes in region R_i using restricted flooding and $t_{refresh}$ is reset.
3. If a node in R_i does not receive a refresh of Q within time interval $t_{refresh}$ it removes all code for this query.

Besides this periodical processing GHT requires the following background processing: Perimeter Refresh Protocol and the GPSR update protocol. To remove a query from the sensor network, only the home node of an area must be informed. The query removes itself after period $t_{refresh}$ has elapsed regardless of topology changes and node failures.

4 Conclusion

The contribution of this paper is a hybrid data-dissemination technique providing a data-centric view. Compared with other algorithms, it has the advantage of using the energy of the individual nodes of the sensor network more uniformly, and compared to pure in-network aggregation the algorithm provides a higher resistance against node failures due to the replication of data on the nodes of the home perimeter. Currently the algorithm is empirically evaluated and compared to other approaches using the simulation tool ns-2.

References

[GEG$^+$03] Greenstein, B., Estrin, D., Govindan, R., Ratnasamy, S., und Shenker, S.: DIFS: A distributed index for features in sensor networks. In: *Proc. First IEEE Int. Workshop on Sensor Network Protocols and Applications*. S. 163–173. IEEE Press. 2003.

[GGC03] Ghose, A., Grossklags, J., und Chuang, J.: Resilient Data-Centric Storage in Wireless Ad-Hoc Sensor Networks. In: *Mobile Data Management - MDM 2003*. S. 45–62. 2003.

[IGE00] Intanagonwiwat, C., Govindan, R., und Estrin, D.: Directed diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In: *Proc. of the 6th annual int. conf. on Mobile computing and networking*. S. 56–67. ACM Press. 2000.

[KK00] Karp, B. und Kung, H. T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: *Mobile Computing and Networking*. S. 243–254. 2000.

[MFHH02] Madden, S., Franklin, M., Hellerstein, J. M., und Hong, W.: TAG: a Tiny AGgregation service for ad-hoc sensor networks. *SIGOPS Oper. Syst. Rev.* 36(SI):131–146. 2002.

[RKS$^+$03] Ratnasamy, S., Karp, B., Shenker, S., Estrin, D., Govindan, R., Yin, L., und Yu, F.: Data-centric Storage in Sensornets with GHT, a Geographic Hash Table. *Mob. Netw. Appl.* 8(4):427–442. 2003.