A PUBLISH/SUBSCRIBE PROTOCOL FOR RESOURCE-AWARENESS IN WIRELESS SENSOR NETWORKS

Salman Taherian and Jean Bacon
Computer Laboratory
University of Cambridge, UK
{Salman.Taherian,Jean.Bacon}@cl.cam.ac.uk

Abstract Research has shown that alternating the routes in wireless sensor networks, can extend the network's lifetime and increase communication savings. Many proposed data dissemination protocols, however, establish paths that guide data on a hop-level basis from sources to sinks. In this paper, we explain why such a tight-coupling (between the data dissemination and data routing) may be desired, even when a location-based addressing scheme may be available; and describe a novel location-based publish/subscribe protocol that offers support for the transparent operation of resource-aware routing protocols. Our protocol offers a trade-off between shared event dissemination paths (that can increase communication efficiency) and support for resource-awareness (that allows freedom of notification routing from the publishers to the subscribers).

Keywords: publish/subscribe, location-based, resource-awareness, sensor networks

1. Introduction

In wireless sensor networks (WSNs), where nodes are power-constrained, prolonged operation and the network's survival are desired. Research [13] has shown that alternating routes, when delivering data from sources to sinks, can extend the network's lifetime by distributing the communications load within the network. Fixed paths, however, allow the formation of shared data dissemination links, over which data (and hence the communication costs) are shared among multiple sinks.

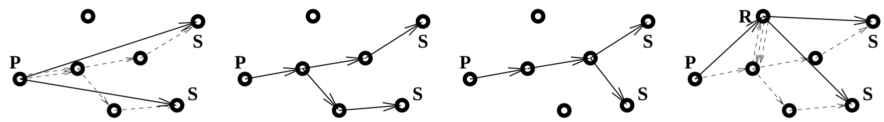
In this paper, we present Quad-PubSub, a publish/subscribe protocol for location-aware WSNs. Quad-PubSub decouples itself from the routing layer to enable the transparent operation of resource-aware routing protocols, [13][4][15], that can extend the network's lifetime. We label this contribution as “support for resource-awareness” and measure it through the level of routing restrictions that Quad-PubSub imposes over the routing of events from publishers to subscribers. We also construct shared event dissemination links (referred to as “shared paths”) that reduce the overall communications cost. This enhances scalability, but also conflicts with the first goal (discussed later in section 3). We use a subscriber-given ϵ factor to manipulate this trade-off, and have developed a localized algorithm which resolves the subscribe and unsubscribe operations efficiently within the network. Quad-PubSub establishes links without the involvement of end-point publishers or subscribers, meaning that we do not rely on acknowledgments or re-enforcements. It is a publish/subscribe protocol in that it decouples publishers and subscribers through a set of intermediate nodes, called event brokers.

2. Related Work

There have been many data dissemination protocols, designed for WSNs. Perhaps, one that is most well-known is directed diffusion[8] and its family of protocols [14]. These data-centric protocols perform combined data dissemination and routing, in that data is delivered from sources to sinks in the absence of an external routing protocol. Shared paths can be formed; cheaply, along the lowest latency paths (opportunistic sharing); or globally (greedy sharing), to reduce the communications cost [10][7]. Where geographical information is available, researchers have used this ([16][14][6]) to reduce the path establishment costs, but have still maintained a tight-coupling between data dissemination and routing. This tight-coupling helps to efficiently construct shared paths (detailed more in the next section). SAFE[9] is a data-centric protocol, that uses tight-coupling to form shared paths in location-aware WSNs.

Other works, however, have shown that resource-awareness takes a more dominant role in extending the operational lifetime and network's survival. Shah et al. [13] have shown that a resource-aware approach can extend the energy savings by 21.5% and the network's life-time by 44%, when compared to the optimal paths used in directed diffusion. They maintain a set of sub-optimal paths, chosen by means of a probability function, from which they select a single path randomly to deliver data.

We use the location information to decouple our protocol from the routing mechanism, thereby allowing the transparent operation of these protocols and the like ([13][4][15]). Nonetheless, we also address the challenge of forming shared paths efficiently and effectively to reduce the communication costs (when optimal routes are used).



(a) Direct Unicast (b) Opportunistic (c) Greedy Sharing (d) GHT-based

Figure 1. Event forwarding paths for two subscribers and a single publisher (solid lines are pub/sub links, dashed lines are shortest routes)

Our work resembles GHT[12], DIMENSIONS[3], DIMS[11], and DIFS [5], in that it uses a geographical hash function. These works do not support information dissemination, but address a data storage and search problem in sensor networks. We are not aware of any information dissemination protocol that makes use of these approaches, nevertheless we have examined a comparable GHT-based approach in our evaluations.

3. Preliminaries

In systems where geographic information is available, a globally unique addressing scheme may be trivially set up. The event dissemination protocols may use these identifiers to decouple themselves from the underlying routing, which allows the transparent operation of external routing protocols. Shared paths, however, offer worthwhile savings when coupling is tighter. Let us consider figure 1 as a case study.

Direct unicast links can be set up, see figure 1(a). In this setup, maximum support for resource-awareness is achieved by having the subscribers' details only at the publisher's node. Event notifications can take any route from the publisher to the subscribers, guided by the underlying routing protocol. The lowest notification delivery cost (assumed proportional to the number of hops taken) is 6 hops.

Figure 1(b) shows an equivalent case where opportunistic shared paths are formed. States are stored at every individual hop, and used to merge overlapping lowest latency paths. The delivery cost is now 5 hops. Interestingly, these savings are made at the publisher's region which help to extend the life-time of the publisher's and surrounding nodes. A tight-coupling, however, has already been formed, that compromises the support for resource-awareness. At the expense of higher communication costs (e.g. network broadcast), more effective information-sharing paths can be formed, see figure 1(c). The notification delivery cost is now 4 hops, but the subscription handling mechanism has a cost proportional to that of network broadcast. Other downsides of this tight-coupling are high exposure of the protocol to low-level failures and dynamics, and possible dependencies over environmental characteristics, such as symmetric link connectivity for reverse-path routing.

Event Topics E	Publi–shers geo−id	Subscribers geo−id, coverage, epsilon
Temp	<local>	<25,12>, {[0,0]−[20,20]}, 1.5
		<local>, {[22,7]−[30,22]}, 1.75

Figure 2. ECT

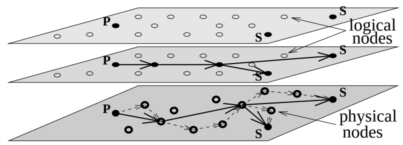


Figure 3. Quad-PubSub

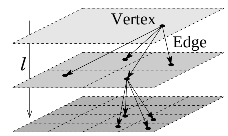


Figure 4. GSs

Rendezvous-based approaches can offer a trivial solution to this tight-coupling problem. Using GHT[12], one can construct shared paths while supporting resource-awareness. In this setup, subscriptions are joined and shared paths are set up at the defined rendezvous nodes, see figure 1(d). Support for resource-awareness is better than the opportunistic and greedy approaches, but worse than direct unicast as events must still route through the defined rendezvous nodes. This approach, however, has a number of disadvantages that significantly outweigh its benefits. Firstly, the rendezvous nodes are selected statically and according to their locations, as opposed to their level of resources. Secondly, they are subject to high event handling and dissemination costs that result in their immediate power depletion and failure. Thirdly, the resultant event dissemination paths (even in the case of shortest-distance routing) result in high communication costs; the cost is 8 hops in figure 1(d).

4. Quad-PubSub

In Quad-PubSub, information is assumed to have notions of type and location assigned to it. Event notifications have *topic names* and *space* attribute parameters that describe the context and location of events' occurrences. Clients can express interests over event topics and spatial regions, from which they receive event notifications. In this paper, we focus on events, whose space attribute is tied to their publisher's location (i.e. publishers generate events that relate to their immediate locality). The following operations are supported at our protocol's interface.

- *Subscribe (Event_Topic, Coverage_Domain, Epsilon_Factor, Event_Handler)*
- *Unsubscribe (Event_Topic, Coverage_Domain)*
- *Publish (Event_Topic, Space_Attribute, Event_Payload)*

Spatial regions are described in terms of *Coverage Domains*, and the *Epsilon_Factor* is used to construct paths that conform to subscriber's specifications. Event notifications are disseminated by means of using simple *Event Clients Tables (ECTs)*, that are stored at every participating node and direct events to the next set of related nodes in the network (see figure 2). The participating nodes (termed "publish/subscribe entities") comprise *Event Clients (ECs)* and *Event Brokers (EBs)*. ECs are simply the end-point publishers and subscribers. EBs are a set of ordinary nodes, that are selected according to the protocol's preferences and run-time nodal resources. They operate as information-sharing junctions for one or more publish/subscribe entities. Figure 3 shows the set of available EBs at the top layer, and the set of selected EBs at the middle layer. In this example, two EBs are selected. Events are routed through the set of selected EBs to reach the end-point subscribers.

The selection of EBs is important. The number of selected EBs has a direct impact on the discussed support for resource-awareness and the effectiveness of shared paths in the network. The smaller the number, the lower is (a) the routing restrictions imposed over the delivery of events (higher support for resource-awareness), (b) the number of ECT entries stored in the network, and (c) the susceptibility of Quad-PubSub to node failures. Nevertheless, in forming shared paths the higher the number of selected EBs, the higher is the likelihood of a selected event broker to exist at an optimal information-sharing junction, as achieved by greedy sharing (illustrated earlier in figure 1(c)).

Our protocol comprises two operating layers, a *logical layer* and a *physical layer*. The logical layer defines the set of event brokers that exist over an overlay layer (the top layer in figure 3). It defines the relationship that event topics, location and event brokers have with one-another, and includes a localized resolving algorithm which selects EBs for event dissemination (the middle layer in figure 3). These operations are decentralized and subscription-driven. The physical layer reflects the protocol's knowledge of the underlying network and nodal characteristics. It is responsible for the resource-aware operation of the protocol, and provision of network-related services, such as that of fault-tolerance. The bottom layer in figure 3 shows this physical layer, in which the two selected EBs are mapped to two nodes in the physical network, and publish/subscribe links are formed over the real infrastructure. The next section describes our system formally.

4.1 Notation

The entire network's coverage area is enclosed in a region, referred to as the network space S. The logical layer is described by a graph, $G_{E,L} = (G_{E,LV}, G_{E,LE})$, whose vertices $G_{E,LV}$ define all the logical EBs, relating to the set of publishable event topics E, and directed edges $G_{E,LE}$ describe the EBs' parent-child relationships (discussed later). The real network is described by the physical layer graph $G_P = (G_{PV}, G_{PE})$, whose vertices G_{PV} represent the deployed network nodes, and directed edges G_{PE} describe the link-layer connections between them.

- $loc(u \in (G_{E,LV} \cup G_{PV})) \mapsto s \in S$ is a function that maps a vertex u to a point, s, on the network space S. $l(u \in G_{E,LV}) \equiv loc(u)$ and $p(v \in G_{PV}) \equiv loc(v)$ are short-hand notations.
- $hash_{e \in E}(s \subseteq S) \mapsto p \in s \subseteq S$ denotes a geographical hash function, that when given a key (event topic e) and a spatial region s, outputs a unique geographical coordinate p within the given region, s.
- $map(u \in G_{E,LV}) \mapsto v \in G_{PV}$ denotes a one-way mapping function, that maps every vertex on the logical layer to a vertex on the physical layer. The function is a resource-aware mapping function, implemented by the physical layer.
- $res(u \in G_{PV}, v \in G_{E,LV}) \mapsto y \in \{true, false\}$ is a boolean function that returns whether a physical node u has sufficient resources to operate as an information-sharing junction or not.

4.1.1 Logical Layer's Notation. The logical layer partitions S into a hierarchy of geographical scopes, in which each scope is subdivided into four equisized geographical scopes (GSs), see figure 4. The geographical scopes are static, and total to a number of $\frac{4^l-1}{3}$ scopes for an l level hierarchy. For every combination of an event topic $e \in E$, and a geographical scope $g \subseteq S$, an event broker $u \in G_{e,LV}$ is defined that is responsible for events matching the event topic e and holding space attribute $s \in g$. If one interconnects the EBs related to the event topic e, from the highest geographical scope to the lowest geographical scopes, a *quad tree (QT)* is formed, where every vertex has four child vertices (see figure 4). Since the operation of the logical layer is independent of the event topics, we study our protocol from the perspective of a single event topic, $e \in E$, and its corresponding QT, $G_{e \in E,L} \equiv G_L = (G_{LV}, G_{LE})$.

- $u \in G_{LV}$ denotes a vertex that represents an EB on the QT.
- $(u, v) \in G_{LE}$ denotes a directed edge that represents the parent-child relationship between u and v on the QT (see figure 4).
- $c(u \in G_{LV}) = \{v \in G_{LV} | (u, v) \in G_{LE}\}$ denotes the children of u.
- $s(u \in G_{LV}) \mapsto s \subseteq S$ is a function that returns the geographical scope for vertex u. Also, $s_{i \in \{1, \cdots, 4\}}(u \in G_{LV}) \equiv s(c_i(u))$.
- $c_{i \in \{1, \cdots, 4\}}(u \in G_{LV})$ denotes the ith child of u, that is located in the ith sub-geographical scope of $s(u)$, starting from the quadrant with the minimum coordinate values and counting clock-wise.
- $cov^{v \in G_{PV}}(u \in G_{LV}) \mapsto s \subseteq s(u)$ is a function that returns the run-time coverage of u, on behalf of its interconnected subscriber v, over its defined scope, $s(u)$. Also, $cov(u \in G_{LV}) \equiv \bigcup_{v \in G_{PV}} cov^v(u)$.

- $R(u \in G_{LV}, p \in G_{PV}, s \subseteq S, \epsilon \in \mathbb{R}) \mapsto \{(v \in G_{LV}, s_v \subseteq S, \epsilon_v \in \mathbb{R}) | v \in c(u) \cup \{u\}, s_v \subseteq s(v), \epsilon_v \in \mathbb{R}\}$ is a *Localized Resolving Algorithm* that when given a logical vertex u, subscriber p, and a coverage domain $s \subseteq s(u)$, returns a set of triples that describe the logical vertices and coverages for which the vertices are responsible.
- $K(p \in G_{PV}, q \subseteq S) \mapsto \{u \in G_{LV} | \forall v \in K(p,q)\ cov^p(u) \cap cov^p(v) = \emptyset, \bigcup_{x \in K(p,q)} cov^p(x) = q\}$ describes the overall operation of the logical layer, in which a subscription from p with coverage domain q is fully resolved over the QT. An EB $u \in G_{LV}$ is selected if $u \in K(p,q)$.

4.2 Logical Quad-Trees Layer

Event brokers offer access to event notifications that fall within their defined geographical scopes. They initially receive no event notifications, and then subscribe or unsubscribe independently to meet their interconnected pub/sub entities' requirements. $l(u \in G_{LV}) = hash(s(u))$ statically defines the location mapping of every EB on S. Subscribers are linked to those event brokers that can serve them with their set of interested events. Since geographical scopes overlap in space, EBs also overlap in coverage scopes. Thus, there exists not just one, but at least $2^l - 1$ different combinations of event brokers that can satisfy a subscription request (l, here, is the depth of the geographical scope hierarchy).

The logical layer selects a number of EBs, $K(p,q)$, which have ECT entries that forward events to the subscriber p with coverage domain q. These ECT entries (for p) are reflected by $\{cov^p(u) | u \in K(p,q)\}$. The registered coverages, for p, at the selected EBs are always mutually exclusive to ensure *exactly-once delivery* ($\forall u, v \in K(p,q)\, u \neq v$, $cov^p(u) \cap cov^p(v) = \emptyset$), and sum to the subscriber's coverage domain of interest, q, to ensure *correctness* ($\bigcup_{x \in K(p,q)} cov^p(x) = q$), see figure 5.

In order to determine $K(p,q)$, in a decentralized manner, we have developed a *localized resolving algorithm*, $R(u, p, s, \epsilon)$, which iteratively resolves the (un)subscribe operation over the QT. A subscribe (or unsubscribe) operation (at the EC) is packaged into a subscription (or unsubscribe) message, and dispatched to the *nearest covering event broker*, from which the iterative operation starts and continues down the QT until complete resolution. The nearest covering event broker to a subscriber $a \in G_{PV}$ with coverage domain $q \subseteq S$ is a $u \in G_{LV} : q \subseteq s(u)$, such that also $\forall v \in G_{LV} : q \subseteq s(v), |p(a) - l(u)| \leq |p(a) - l(v)|$.

The R algorithm determines the set of responsible EBs, for a subscribe or unsubscribe operation, based on the event dissemination path lengths and registered coverage domains at the local EB. Its output comprises

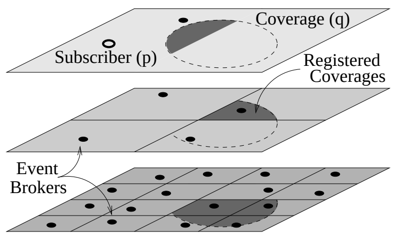


Figure 5. Resolved Subscription

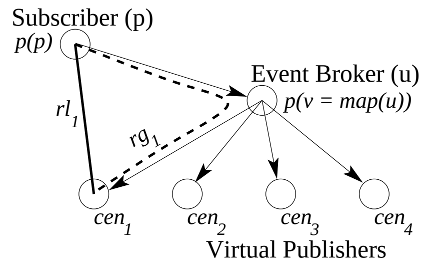


Figure 6. Register vs Relay distances

a set of triplets, that contain the set of event brokers and the set of coverage domains for which the EBs are held responsible.

If a node $u \in G_{LV}$ is contained within its R's output, $(u, s', \epsilon') \in R(u, p, s, \epsilon)$, then u is *selected* for the event notification forwarding (or ECT deletion) and $u \in K(p, q)$. This selection triggers a *registration* (or *de-registration*) process that is detailed shortly. If the set of u's children are involved in the function's output, $(v \in c(u), s', \epsilon) \in R(u, p, s, \epsilon)$, then the operation is said to have been *decomposed and relayed*. The original message is then forwarded to v, with modified coverage domain s'.

Figure 5 shows an example, in which a subscriber's subscription is forwarded to the highest level EB (shown at the top layer). The subscription coverage domain, q, is partially registered at the top level EB (shown as a shaded region), and partially decomposed and relayed to the lower layer EBs, which correspond to the unshaded region of q. The process is iterated until the subscription coverage domain, q, is completely resolved over the QT. Selected EBs are those which are responsible for the registered (shaded) areas. In this example, six EBs are selected (one at the top layer, one at the middle layer, and four at the bottom layer).

4.2.1 Localized Resolving Algorithm.

This section describes $R(u, p, s, \epsilon)$, where $u \in G_{LV}$ is the evaluating event broker, $p \in G_{PV}$ is the subscriber, $s \subseteq S$ is the coverage domain for which u is held responsible, and ϵ is a guidance factor that is specific to the subscription operation and is discussed shortly.

We first discuss the unsubscribe operation. The R function for this operation can be formally expressed as $R(u, p, s, \epsilon) = \{(u, s \cap cov^p(u), \epsilon)\} \cup \{(v \in c(u), r \cap s(v), \epsilon) | r = s - cov^p(u)\}$. This means that u de-registers the overlapping coverage domain that it had previously registered for p and decomposes and relays the remainder to its child vertices (EBs).

For a subscription operation, the algorithm analyses the coverage domain s in relation to the already covered domain $cov(u)$. It computes the impact of its involvement (*register*) or deferral (*relay*) on the resulting event forwarding path, and evaluates that against a given ϵ factor.

The subscriber-given $\epsilon \in \mathbb{R} : \epsilon \geq 1$ indicates the ratio of the longest event forwarding path that the subscriber $p \in G_{PV}$ is satisfied with, relative to the theoretically shortest event forwarding path. The ϵ factor empowers p to control the following properties of the formed event forwarding path. A lower ϵ value would shorten the event forwarding path, such that the lower bound of *notification delivery latency* is reduced. A lower ϵ value would also decrease the number of selected event brokers, thus maximising the *support for resource-awareness*. A larger value, however, promotes higher *path-sharing*, such that the path is stretched to increase overlap among multiple paths.

u computes approximate interconnection distances for the options of registering or relaying the subscription, with respect to each of its sub-coverage scopes $s_i(u)$. $\{rg_i \in \mathbb{R} | i \in \{1, \cdots, 4\}\}$ and $\{rl_i \in \mathbb{R} | i \in \{1, \cdots, 4\}\}$ denote the sets of distances for these two options, see figure 6. The algorithm uses p's and u's real physical locations, $p(p)$ and $p(v = map(u))$, in its computations, and computes four virtual publishers' coordinates that reflect the publishers in each of its sub-geographical scopes. The four latter coordinates are defined as the *centroid points*, $\{cen_i \in S | i \in \{1, \cdots, 4\}\}$, of uncovered spatial regions $\{sr_{i \in \{1, \cdots, 4\}} \subseteq S | sr_i = s_i(u) \cap (s - cov(u))\}$. The distances are then computed as $\{rl_i \in \mathbb{R} | rl_i = |p(p) - cen_i|\}$ and $\{rg_i \in \mathbb{R} |$ if $sr_i = \emptyset$, then $rg_i = |p(p) - p(v)|$, otherwise $rg_i = |p(p) - p(v)| + |p(v) - cen_i|\}$.

For every $i \in \{1, \cdots, 4\}$, where $\frac{rg_i}{rl_i} \leq \epsilon$, the subscription is registered, $R(u, p, s, \epsilon) = R(u, p, s, \epsilon) \cup \{(u, s \cap s_i(u), \epsilon_i) | \epsilon_i = \frac{\epsilon.|p(p) - cen_i| - |p(p) - p(v)|}{|p(v) - cen_i|}\}$. Otherwise, the subscription is relayed, $R(u, p, s, \epsilon) = R(u, p, s, \epsilon) \cup \{(c_i(u), s \cap s_i(u), \epsilon)\}$. Where u is a leaf node on the quad-tree (i.e. $c(u) = \emptyset$), then the subscription is fully registered, $R(u, p, s, \epsilon) = \{(u, s, 0)\}$.

4.2.2 Register/de-Register.

Registering a subscription coverage $s \subseteq S$, for a subscriber $p \in G_{PV}$ at node $u \in G_{PV}$, modifies u's ECT to reflect $cov^p(u) \supseteq s$. A de-registration process operates similarly. It removes an unsubscribe coverage s from the existing coverage $cov^p(u)$.

If u is an event broker, register and de-register operations may trigger subsequent independent subscribe or unsubscribe operations by u itself. When u registers an $s \subseteq S$, such that $s \nsubseteq cov(v)$, then u initiates a subscribe operation for the remainder coverage, $s - cov(u)$, with an ϵ given by R. Similarly, when $cov(u) = \bigcup_{v \in G_{PV}} cov^v(u)$ is affected, after a de-registration process, u unsubscribes the excess coverage domain, $cov_{previous}(u) - cov(u)$. These operations differ from the subscribe/unsubscribe operations initiated by ECs in two ways. Firstly, the messages are strictly forwarded down the QT, to u's children, $c(u)$. Secondly, when u is a leaf node of the quad-tree, the subscribe/unsubscribe messages are broadcast to all the nodes within its defined scope, $s(u)$.

4.3 Physical Publish/Subscribe Layer

The operation of the physical layer is not discussed in this paper. It maps the selected logical event brokers to real physical nodes, such that $map(u \in K(p,q)) \mapsto v \in G_{PV} : res(v,u) = \{true\}$. It implements a *hand-over* procedure which actively relieves v, from its operations, when its resources fall short, $res(v,u) = \{false\}$, and finds a new resource-ful node that can operate instead. Finally, fault-tolerance is supported here, through redundant storage of ECT entries in the network, that deliver events to end-point ECs even when mapped EBs happen to fail.

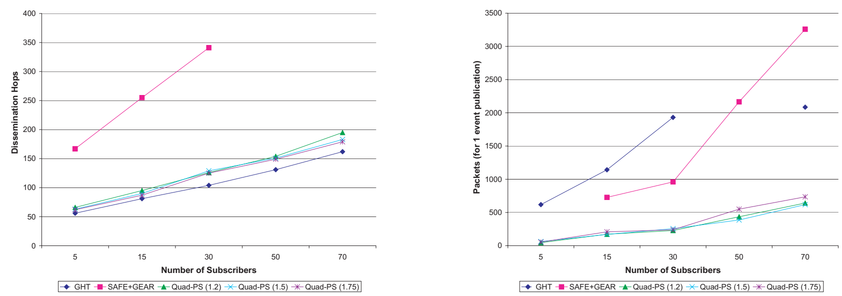
4.4 Event Dissemination (Publish Operation)

Publishers actively introduce events, using the publish operation. Event notifications, $\{(e \in E, s \in S, \texttt{payload})\}$, are distributed to all the subscribers, whose interests match the event topic e and coverages contain s. It is worth noting that the publish operation does not incur *any* communications cost *unless* at least one subscriber has expressed an interest on the published event. The transparency and anonymity of ECs is supported through the involvement of intermediate EBs.

5. Evaluation

We are comparing Quad-PubSub against two classes of protocols, GHT-based and data-centric. The GHT-based approach involves rendezvous nodes at which point subscriptions are joined and shared paths are constructed. For the data-centric approach, we selected SAFE[9], which uses a similar subscription-driven process, and forms shared paths in location-aware WSNs. SAFE, however, assumes that each request is of concern to a single sensor node. In order to support multiple sensors, we augmented SAFE with GEAR[16]. In this combination, subscriptions reach those regions of interest and are then broadcast within the regions. Subscription messages that meet data dissemination paths, prior to reaching those regions, are joined as in [9].

We're examining support for resource-awareness, by means of the number of nodes that notifications *must* route through to reach the subscribers. This is equivalent to the number of hops taken on the logical layer. We're also examining efficiency of the resulting event dissemination paths, by means of the number of messages that are transmitted, to deliver an event from the publishers to the subscribers.



(a) Hops vs Subscribers

(b) Messages vs Subscribers

Figure 7. Preliminary results (800 nodes in a 256x256 grid). Quad-PubSub was evaluated with $\epsilon = \{1.2, 1.5, 1.75\}$.

We're using the discrete event simulator JiST/SWANS[1], with wireless parameters from CC1000 radio[2]. Results are still being analyzed. Nevertheless, some early results (see figure 7) suggest the following.

- Combination of SAFE and GEAR perform very poorly, for two reasons. Firstly, the subscription messages may enter the region of interest from different points, which result in formation of multiple overlapping event dissemination trees in the region. Secondly, effective opportunistic sharing is rarely achieved in dense networks. This is due to the greedy forwarding policy of many geographical routing protocols, which decreases the chances of intersection.
- GHT protocol's performance varies significantly according to the outcome of the hash function and the subscriber's region of interest. We have yet to determine a formal way of comparing this randomness against our steadily well-performing protocol.
- The impact of the *Epsilon_Factor* is least noticeable in figure 7, as we used a fixed subscription coverage domain. This promotes the formation of shared event dissemination paths among the subscribers.

6. Future Work & Conclusions

In this paper, we presented a publish/subscribe protocol that allows the transparent operation of resource-aware routing protocols which can extend the WSN's lifetime. In conjunction with this goal, we targeted shared event dissemination paths that can lower communication costs. Quad-PubSub contains a localized resolving algorithm, which iteratively resolves the subscribe/unsubscribe operations over the network. The algorithm is simple in operation, comprising distance calculations, but early results suggest that it can offer significant savings in WSNs. This work is still under evaluation, and in future will be supported by formal proofs, detailed analytical and experimental evaluations.

References

[1] Rimon Barr, Zygmunt J. Haas, and Robbert van Renesse. Scalable wireless ad hoc network simulation. *Handbook on Theoretical and Algorithmic Aspect of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks*, pages 297–311, 2005.

[2] CC1000 single chip very low power rf transceiver. http://www.chipcon.com/files/CC1000_Data_Sheet_2_1.pdf.

[3] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: Why do we need a new data handling architecture for sensor networks? *ACM SIGCOMM Computer Communication Review*, 33(1):143–148, 2003.

[4] O. Ghica, G. Trajcevski, , and P. Scheuermann. Alternating routes in sensor networks (extended abstract). In *In Proc. Int. Conf. on Information Processing in Sensor Networks (IPSN) - Work In Progress Session*, Nashville, USA, 2006.

[5] B. Greenstein, D. Estrin, R.Govindan, S. Ratnasamy, and S.Shenker. DIFS: A distributed index for features in sensor networks. In *Proc. 1st IEEE Intl. Workshop on Sensor Network Protocols and Applications (SNPA)*, Anchorage, AK, May 2003.

[6] John Heidemann, Fabio Silva, and Deborah Estrin. Matching data dissemination algorithms to application requirements. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 218–229, New York, NY, USA, 2003. ACM Press.

[7] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heideman. Impact of network density on data aggregation in wireless sensor networks. Technical Report 01-750, U. of Southern California, Computer Science Department, 2001.

[8] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.

[9] S. Kim, S.H. Son, J.A. Stankovic, S. Li, and Y. Choi. SAFE: A data dissemination protocol for periodic updates in sensor networks. In *Proc. Workshop of IEEE Intl. Conf. on Distributed Computing Systems (ICDCS)*, Providence, RI, March 2003.

[10] B. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Proc. Workshops of 22nd Intl. Conf. on Distributed Computing Systems*, pages 575–578, Vienna, Austria, July 2002. IEEE Computer Society.

[11] X. Li, Y. J. Kim, R. Govindan, and W. Hong. Multi-dimensional range queries in sensor networks. In *Proc. 1st Intl. Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 63–75, Los Angeles, CA, November 2003. ACM.

[12] Sylvia Ratnasamy, Brad Karp, Li Yin, Fang Yu, Deborah Estrin, Ramesh Govindan, and Scott Shenker. GHT: a geographic hash table for data-centric storage. In *Proceedings of the first ACM international workshop on Wireless sensor networks and applications*, pages 78–87. ACM Press, 2002.

[13] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, March 2002.

[14] Fabio Silva, John Heidemann, Ramesh Govindan, and Deborah Estrin. Directed diffusion. Technical Report ISI-TR-2004-586, USC/Information Sciences Institute, January 2004.

[15] Goce Trajcevski, Oliviu Ghica, and Peter Scheuermann. Car: Controlled adjustment of routes and sensor networks lifetime. In *MDM '06: Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)*, page 23, Washington, DC, USA, 2006. IEEE Computer Society.

[16] Y. Yu, R. Govindan, and D. Estrin. Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks, 2001.