Building an Efficient Overlay for Publish/Subscribe in Wireless Sensor Networks

Claude Chaudet, Nicola Costagliola, Isabelle Demeure, Salma Ktari, Samuel Tardieu

To cite this version:

Claude Chaudet, Nicola Costagliola, Isabelle Demeure, Salma Ktari, Samuel Tardieu. Building an Efficient Overlay for Publish/Subscribe in Wireless Sensor Networks. 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Jun 2012, Seoul, South Korea. pp.362-370, 10.1109/SECON.2012.6275798 . hal-00780006

HAL Id: hal-00780006
https://hal.science/hal-00780006v1

Submitted on 16 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Building an Efficient Overlay for Publish/Subscribe in Wireless Sensor Networks

Claude Chaudet, Nicola Costagliola, Isabelle Demeure, Salma Ktari and Samuel Tardieu
Institut Télécom / Télécom ParisTech
CNRS LTCI UMR 5141
Paris, France
Email: {first.last}@telecom-paristech.fr

Abstract—**In this paper we examine how to efficiently build a brokers overlay to implement publish/subscribe in a wireless sensor network, trying to reduce sensor nodes energy consumption, memory required for buffering packets and delivery time. We evaluate, on the connectivity graph that represents the network, the performance of various criteria that can be used to select brokers among the set of nodes. We compare a dominating set approach with the selection of the best ranked brokers based on centrality measures. We finally give hints on how to implement a distributed algorithm to approximate the most efficient overlays.**

***Keywords*-Wireless Sensor Networks, Publish/Subscribe, brokers placement**

I. INTRODUCTION

Auto-organization and energy preservation are key objectives for Wireless Sensor Networks. Deploying such a network for monitoring an area should require little or no calibration, and the network should perform its duty as long as possible, while necessitating little maintenance. Sensors are also supposed to be numerous and thus relatively cheap devices, which means that they have little computation power and memory space and that they use a low-throughput communication channel. Designing an efficient middleware for such an environment is therefore challenging and requires to combine a set of efficient components.

Concerning communication, the publish/subscribe paradigm represents an interesting building block, as this communication scheme is fully asynchronous. In this paradigm, receivers specify filtering expressions on data descriptors that allows the network to filter irrelevant data before it reaches the destination. The filtering operation that matches receivers interests with data descriptors can be performed anywhere in the network, for example by the sources or by dedicated intermediate nodes, called *brokers*.

In this latter case, information producers and consumers do not need to know each other, but only need to be able to join one such broker. Sensor nodes may send their measurement results to any broker, alongside with a set of data descriptors such as the localization of the measurement, the type of data, etc. Receivers (e.g. control centers and operators) also express their interest to the brokers, which are then in charge of filtering and dispatching the information efficiently within the network. With this scheme, the brokers are the only nodes that need to know and maintain addresses and routing paths. A regular node only needs to maintain a route towards one broker, while the brokers form an overlay to route packets towards their true destinations.

Having too few brokers results in concentrating the traffic into a few areas that will soon become overloaded. Moreover, the brokers in this case support a high load and their batteries levels will decrease quickly. On the other hand, having too many brokers generates a high load of control traffic, as brokers should exchange state information. It also means that data has to be duplicated more than necessary to reach every relevant broker that, in turn forwards it to every interested subscribers. Therefore, the shape of the brokers overlay, i.e. the number and location of the brokers has a significant impact on the network energy consumption, on the information delivery time and on the load of each intermediate node.

There are multiple possibilities to select brokers among the set of nodes. Building a dominating set of the network topology is at the heart of several clustering algorithms and the corresponding algorithms are potentially good candidates for brokers selection, even though clustering is a slightly different problem. Centrality measures, coming from the social networks field, are also appealing metrics, as they reflect the importance of network nodes in a given topology. Moreover, most of the centrality measures can be approximated by using only local information, which make them good candidates for a distributed brokers selection algorithm.

In this paper, we compare the performance of various brokers overlays on two main families of criteria. The first criterion is the overall network energy consumption profile, which depends on the routes that are used in the network and on the network density, as sensor nodes spend energy when emitting frames but also when receiving frames. The second consideration is the sensors load. Memory is very scarce in sensor nodes, and particularly RAM. Therefore, packets buffering is limited and the traffic intensity that passes through every node should be limited. Moreover, the queues size have an immediate effect on delivery times and keeping them under a certain limit can be necessary to meet QoS constraints. If this depends on the routing algorithm, it also depends on the brokers selection, as brokers are the source or the destination of all data flows.

The rest of this paper is organized as follows. Section II presents contributions that are related to the problem of selecting brokers for a publish/subscribe architecture. Section III presents and justifies the energy consumption and load model that allows us to evaluate brokers selection strategies. In section IV we present the various criteria that we compare in section V through simulation of the previous model. section VI concludes the paper and presents future working directions, including the way to build a distributed algorithm that uses these metrics.

II. RELATED WORKS

A great number of publish-subscribe architectures have been proposed for the Internet and several contributions identify that the organization of the brokers overlay is an important issue. SIENA [1] propose to organize the overlay as a tree, JEDI [2] and Rebeca [3] use directed acyclic graphs. Kyra [4] uses an interconnection of cliques and Scribe [5], Bayeux [6], Medghoot [7] rely on a distributed hash table. Several contributions, such as Rebeca [3], use a P2P overlay to organize the brokers network. [8] propose a heuristic to organize the broker overlay for self-optimizing the number of exchanged messages.

This literature shows that the organization of the overlay has an influence on the performance of the publish/subscribe system. However, the context of the Internet is very different from a wireless sensor network. Energy is a primary concern in sensor networks and the resources scarce. Moreover the traffic pattern is fundamentally different, as in the Internet a publish/subscribe system is classically composed of few publishers and many subscribers, while in a wireless sensor network we expect to have many sensors and few collection points.

Ad-hoc networks are closer to sensor networks, as they are also wireless multihop networks. Brokers overlays designed for these networks tend to take into account the routing protocol to build the brokers overlay, reducing the overhead and better mapping publish/subscribe paths to the actual paths in the network. For instance, in Transhumance [9] the broker overlay directly and dynamically maps to the OLSR Multiple Relay Nodes (MPR) graph defined at the routing layer.

Even though such contributions are closer to our goal, they do not focus on energy either. Moreover, they are strongly dependent on the underlying ad-hoc routing protocol that provides direct routes between all couple of nodes, while sensor networks routing protocols tend to organize the network as a tree or a directed acyclic graph, minimizing the number of routes that a node needs to maintain and thus saving energy and memory space. Still, they demonstrate that the shape of the brokers overlay has a significant impact on performance in these mono-tenant wireless networks too.

Surprisingly few contributions have examined the effect of the brokers overlay in wireless sensor networks. Tran and Truong detail in [10] a considerable number of publish/subscribe systems that can be found in the literature. These systems specify the basic mechanism for transmitting publications and subscriptions, for subscription aggregation and for publication matching. However, most of these approaches do not assume anything on the presence or distribution of brokers. They rather specify a particular routing process that makes the event notification to the subscribers possible and efficient. The only exception is MQTT-S [11], which supposes that publication matching is carried out by a single central broker located on a external wired network. Similarly, Messo & Presso [12] organize the system around a single broker located on the traffic sink.

In [13], the network is divided in square cells that group static sensors, while one or more mobile devices subscribe to topics published by these sensors. Each cell has a node which acts as broker and forms an overlay with all the other brokers. Authors of [14] focus their work on the provisioning of a sensor presence service, based on publish/subscribe pattern and a multiple broker architecture. The brokers are connected with each other, and everyone is responsible for a specific domain of publishers/subscribers.

None of these contributions details criteria for selecting the brokers in the network, but they rather focus on how these nodes cooperate to route the event notifications.

[15] present a procedure to select brokers and form an overlay, creating groups of nodes, called *virtual brokers*, which share the brokers workload. This procedure is distributed and it is decomposed in three algorithm executed in sequence. The first algorithm creates k-means clusters of close nodes. The second algorithm determines a centroid node for each cluster that becomes the center of a Voronoï cell. The third algorithm selects a group of nodes around the centroid to become a virtual broker for that cluster. The final goal of this procedure is to decrease the number of hops of intra-partition and inter-partition communication, supposing that shortest paths are provided by the routing protocol.

This work, which is the closest to our problematic, presents similarities with clustering algorithms which have been the topic of an abundant literature and we refer the reader to [16] for a comprehensive survey on clustering algorithms for WSNs. Even though the base clusters formation algorithms such as LCA are totally relevant in our scenario, the objective of clustering presents slight differences with brokers selection that pushes us to revisit at least the brokers selection criteria.

The goal of brokers selection is not to build a structure that has a predefined shape, but rather to select the brokers in order to optimize energy and delay. There is no a priori reason, for example, to limit the zone managed by a broker in terms of number of nodes or depth. There is no reason either to limit the brokers density. This consideration has a potentially strong impact on the metric that is used to select brokers. Clustering algorithms try to only consider topological aspects, while brokers selection should take into account traffic patterns. As data passes through the brokers, traffic patterns will evolve adapting to the brokers overlay changes, which makes the metric more dynamic than in the clustering case, with possible retroaction. Finally, clustering is often supposed to build a full coverage of the network, each node being associated to a cluster head. In publish/subscribe, only publishers and subscribers need to be associated to brokers. For example, nodes that only perform routing, extending coverage, do not need to be associated with any broker.

III. ASSUMPTIONS AND PERFORMANCE CRITERIA

We suppose, in the rest of this article, that we work over a static wireless sensor network, or at least that mobility is slow enough to be transparent to routing, i.e. routing tables are always up-to-date. We place ourselves above the network layer and suppose that the underlying layer provide routes and an efficient medium access procedure. We also suppose that nodes do not perform duty cycling, even though this assumption is easy to leverage.

Brokers selection should be efficient regarding energy consumption, reducing the total energy spent, but also correctly sharing the load. It should also limit the forwarding duty of every node in the network in order to limit buffering and preserve memory. This section reminds and explains the main results of a model of the problem that was published in [17].

A. Energy consumption due to communication

It is generally admitted in the literature that communication is the dominant source of energy consumption. We assume that IEEE 802.15.4 [18] is implemented at the link layer or that the protocol behaves similarly. This protocol uses a random (CSMA) algorithm to access a single RF channel whose data rate is limited to 250kb/s and limits frame size to 127 bytes. It introduces, for every frame transmission an overhead of $T_h = 992\mu s$ to transmit frame preamble and headers. Emitters can request that an acknowledgment is emitted to confirm a frame reception, which requires $T_{ack} = 352\,\mu s$ to be transmitted.

In digital RF communications, energy is spent for emitting frames, but also for receiving frames, as the signal needs to be filtered out form the noise and decoded. Today's wireless chips present energy consumption figures that are almost identical for sending and receiving modes. If we denote by P_{tx} the power (in Watt) consumed when emitting a frame and by P_{rx} the power consumed when receiving a frame. An emitter that sends a frame of length L bits at a data rate D bit/s to one of its neighbors j spends an energy :

$$E_e(L) = P_{tx}.\left(T_h + L/D\right) + P_{rx}.T_{ack}\,.$$

The receiver, j spends a corresponding energy

$$E_r(L) = P_{rx}.\left(T_h + L/D\right) + P_{tx}.T_{ack}$$

to receive the frame an send the acknowledgment. Moreover, as the wireless channel access procedure is random and does not rely on fixed channels or time slots assignment, neighbors of an emitter have to at least receive the frame header and compare the destination address with their own before knowing if they are recipient of the frame or not. Neighbors of an emitter who are not the intended receiver therefore spend at least an energy equal to $E_o = T_h.P_{rx}$ overhearing frames. This figure can be reduced thanks to an efficient duty cycling scheme, implementing such a scheme requires to maintain nodes clocks synchronization, which has a cost.

Based on these figures, we can evaluate the energy cost of the multihop transmission of a single L bits frame. All emitters (i.e. the sender and the successive forwarders) spends an energy $E_e(L)$, all receivers (i.e. the destination and the forwarders) will spend an energy $E_r(L)$ and all the neighbors of any emitter spends an energy E_o. The transmission of a publication to all the interested receivers therefore has a cost for the network that depends on the length and density of the three segments of the route: from the publisher to the brokers overlay, inside the brokers overlay and from the egress brokers to each subscriber. It will therefore be strongly affected by both the routing protocol and the brokers selection.

B. Routing and nodes loads

At the network layer, several sensor networks routing protocols choose to organize the network as a tree or as a directed acyclic graph. RPL [19] is the current candidate routing protocol for low power networks at the IETF and, even though it may be a bit complex for low-end sensor networks, the structure it builds, a DAG, is representative. These destination-oriented structures fit well the scenario in which most nodes are sources of information while only one or few nodes are destinations. Such an organization is also lightweight for the sensor nodes, as they only need to maintain a route towards the tree root. By default, all frames go upwards the tree, reaching the root which is the only node to have a global vision of the network. The root determines a downwards route and inserts this information in the packet as an optional source routing header. An alternate mode, in which all sensor nodes keep the full vision of their sub-tree is far more efficient in terms of paths lengths but requires nodes to maintain downwards routes. In the rest of this paper, when relevant, we suppose that nodes have the vision of their subtree.

As mentioned in the previous section, the selected routes have an influence on energy consumption, as the longer and the denser the paths are, the more nodes will consume energy. However, when multiple routes are available, aiming for the shortest routes may not be a viable solution either, as the wireless channel on these routes may soon be overloaded as well as the forwarding nodes memory.

To evaluate the traffic load at each node, we make the assumption that each sensor p emitted publications according to a Poisson process of intensity λ_p. Each of these publications is then routed towards the brokers overlay, reaching the broker that is the closest to the emitting publisher. This broker then has to handle a total publications traffic which depends on the number of attached publishers. We then suppose optimistically that all brokers had a clear vision of where the different subscribers lied. This means that brokers should exchange with each other their full subscriptions tables, which is only possible without a noticeable cost if the overall number of subscription is low and if the subscriptions are stable. In a sensor network, this situation is likely to happen, otherwise a synchronization cost within the brokers overlay cannot be neglected anymore. The consequence is that brokers receiving publications know where in the overlay they need to forward these packets. There is therefore inside the overlay a forwarding traffic to bring publications to the brokers where subscribers are attached. These brokers then transmit the event notifications (i.e packets that correspond to publications) to the subscribers. One should note that ingress and egress brokers have the opportunity to compress data and to reduce the data flow, meaning that the traffic intensity that arrives at a subscriber may not be the sum of the traffic intensities of the relevant publishers. For this first evaluation, we supposed that no such compression happened.

From these figures, it is possible to calculate the traffic intensity that every node in the network – publisher, subscriber, broker or simple router – shall support. We can consider that the corresponding service time is defined as the time necessary to emit the frame at the MAC layer, with an average of $1/\mu$ seconds and a variance of σ, taking into account the random backoff and the channel access time. Each node in the network can then be modeled as an M/G/1 queue and using the Pollaczek-Khinchin formula, the mean number of packets in a node's buffer can be expressed as $Q(i) = (1 + (\sigma.\mu)^2).\rho_i^2/\left(2.(1-\rho_i)\right)$, where ρ_i is its load, i.e. the total traffic intensity that is receives divided by the average time required to emit a single frame. For a more complete, multi-channel and comprehensive model, the reader can refer to [17].

This queueing model allows us to evaluate the maximum load that the network may sustain. By limiting the maximum queue size, which reflects the maximum amount of memory that a node may dedicate to packets forwarding, we calculate a maximum load at each node that limits the maximum traffic intensity that a publisher may transmit.

These queues results can also be used to evaluate the time required to cross each forwarding node, and subsequently the end-to-end delivery time of a publication. Applications that need fast delivery can then fix a limit on this end-to-end transmission time, which can then be converted into a maximum nodes load, also limiting the maximum traffic intensity that a publisher may transmit.

IV. BROKERS SELECTION CRITERIA

Based on the performance criteria expressed in the previous section, we compare in this section the performance of various brokers selection criteria.

The first natural structure that could be a good candidate, as it is widely used in clustering, is the **dominating set** over the connectivity graph. A dominating set of a graph $G = (V, E)$ is a subset of the vertices, $V' \subseteq V$ such that each node is either a member of the dominating set, or a direct neighbor of a vertex that belongs to the dominating set. $\forall u \in V, (u \in V')$ or $(\exists v \in V'$ and $(u, v) \in E)$. Various distributed algorithms were proposed to compute such a structure and [20], for example, proves that a dominating set can be built in by a distributed algorithm in $O(\log |V| \log \Delta)$ rounds, where Δ is the maximum degree in the graph, leading to a dominating set larger than the minimum dominating set by a factor $O(\log \Delta)$. Therefore locating the brokers at the vertices that belong to such a dominating set could be an efficient strategy.

However, dominating sets only consider topological aspects and treat all the nodes similarly, regardless of their role (publisher, subscriber or simple routers). Widely used in social networks analysis, centrality measures are also good candidates to build a brokers overlay, as they reflect the importance of nodes regarding various topological and routes-related metrics. There are several ways to measure centrality. The three most widely used centrality measures are Freeman's degree, closeness, and betweenness measures [21]. We may imagine that a centralized algorithm selects n brokers, locating brokers at the n nodes that have the best centrality scores. The correct value for n depends on the scenario.

Degree is a first an easy to obtain centrality measure. It is simply defined as the number of neighbors of a given node an can be obtained by examining broadcasted control packets, e.g. from the routing protocol. High degree nodes are expected to be important in the network, linking several other nodes together. However, in our case, high degree nodes may be bad candidates too because one of their emissions will provoke overhearing at many neighbors.

The **clustering coefficient** can be seen as an extension of the degree. It is defined as the ratio between the number of links that exist between a node's neighbors and the number of links that *may* exist between the same set of neighbors in a complete graph. It is maximum when a node is at the center of a clique. This measure is also easy to obtain through control messages and even if it is not a centrality measure, nodes that have a high clustering coefficient are expected to be well connected and to be able to be reached though relatively short paths.

Closeness centrality measures the average distance between a given node and all the other nodes in the network. Closeness centrality can be regarded as a measure of how long it will take information to spread from a given node to other nodes in the network. It depends on the routing protocol and may be computed in a distributed way by looking at the routing tables. If the routing tables do not contain the full set of nodes, which is likely in a tree-organized sensor networks, they shall contain the distance to the root of the tree and allow to evaluate the distance to the leaves, which will give an indication on the depth of the node in the routing tree.

Finally, **betweenness centrality** is defined by identifying, for every couple of nodes (s, d) in the graph, all the shortest paths between s and d. The betweenness centrality measure of a given node is defined as the number of shortest paths that pass through this node for all the (s, d) couples, divided by the total number of shortest paths between the (s, d) couples. In other words, it represents the proportion of shortest paths in the graph that a node lies on. To take into account traffic, we can restrict the calculation to only take into account active paths, i.e. paths that link a publisher and a subscriber.

Marsden [22] discovered empirically that the egocentric betweenness values, i.e. the betweenness values calculated over a 1-hop neighborhood, had a strong positive correlation to global (called sociocentric) betweenness values for many different network examples. Everett and Borgatti [23] also arrive to a similar conclusion that the two metrics are strongly correlated for most networks. These results show that a distributed computation of betweenness centrality should lead to good results.

Betweenness centrality has already been used and adapted to the sensor networks scenario. Cuzzocrea et al. propose in [24] a weighted bidirectional topology algorithm, called Edge Betweenness Centrality (EBC). EBC selects logical neighbors of a node, so that for each node the set of logical neighbors covers the two-hop node neighborhood. In addition, logical neighbors present the betweenness centrality among the set of 1-hop neighbors. Authors in [25] propose a new topological metric called Sink Betweenness (SBet). This metric considers only the shortest paths that include the sink as terminal node. The main idea is that nodes will choose the next hop among their neighbors that are closer to the sink (in hops) and present the highest SBet.

V. EVALUATION RESULTS

A. Simulation Setup

In order to compare the previous brokers selection criteria, we built a small simulator in C++ that calculates the values of the performance criteria expressed in section III over random networks scenario with a global vision of the network (i.e. in a centralized way). The networks are random geometric graphs, nodes are spread randomly over a square area and an edge exists between a couple of nodes if and only if their distance is lower than a fixed transmission range. Such graphs are similar to unit-disk graphs. We can influence the network density and diameter by letting this parameter vary.

In our study, we fixed the number of nodes in the network to 100, and let the number of publishers and the number of subscribers vary. For each scenario, we ran 100 simulations and calculated the average of the following performance metrics:

- **Energy necessary for one publication round**: based on the energy consumption model described in section III, we evaluate the energy that every node in the network spends for one publication round. Every publisher sends one message that is transmitted to its nearest broker. Brokers then exchange messages so that every broker that serves an interested subscriber gets the whole set of publications. Finally, publications are transmitted to the subscribers by their broker. For this phase, all sending, receiving and overhearing nodes count the energy they spend and the resulting average energy consumption is computed, as well as the fairness of the energy consumption, expressed through Jain's fairness index [26].
- **Maximum load**: based on the M/G/1 queueing model that we evoked in section III, we evaluate the maximum load that the network can sustain. To this end, we first suppose that every publisher emits publication at a frequency of $1\,Hz$. We calculate the resulting queues load in the whole network and then identify the bottleneck(s) node(s), i.e. the nodes that receive the largest traffic intensity. From this maximum intensity, we calculate a scaling factor so that the load of this node reaches a predefined value, arbitrarily fixed to 0.9 in this article. This threshold can derive from a constraint expressed on the maximum buffer space in nodes, or on the maximum end-to-end delivery time, as mentioned in section III
- **Queues sizes and delivery time**: once the maximum load is fixed, we can calculate the queues size and messages delivery time over the scenario.

Concerning routing, we evaluated three situations: in the first situation, we supposed that a routing protocol is able to provide every node the shortest path towards every other node. This scenario allows us to evaluate the effect of the tree-like routing structure. Then we evaluate scenarios in which the network is organized as an RPL-like tree. For RPL, we compared various localizations for the RPL tree root and, as results do not vary much, we only consider here the case in which the root is in the center of the area.

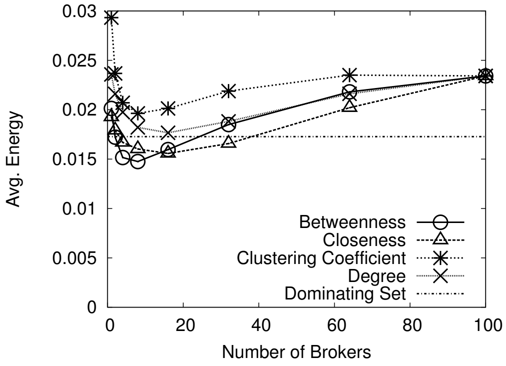
B. Many publishers, few subscribers

We first evaluate the performance criteria in a scenario in which we have a large proportion of publishers (90 nodes in a 100 nodes network) and a reasonable amount of subscribers (20 subscribers), placed randomly. Figure 1(a) represents the average energy spent by all the nodes (expressed in W.s), when the 1, 2, 4, 8, 16, 32, 64 and 100 best nodes according to the different criteria are selected as brokers. In this scenario, routes are the true shortest paths. The dominating set is composed of a fixed number of nodes, which does not vary. Its performance is represented by an horizontal line in all the figures.

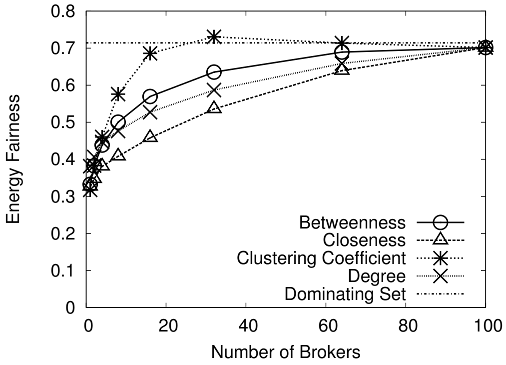
From these figures, we see that the energy is minimal for a relatively low number of brokers. Increasing the number of brokers beyond a certain limit is counter-productive in this scenario, as it introduces unnecessary data replication. The dominating set exhibits a very good performance, even though betweenness centrality and closeness manage to get a better performance when the correct number of brokers is selected. Degree and clustering coefficients are not very effective, as they tend to direct the traffic towards dense areas.

Fairness of the energy consumption is represented on Figure 1(b). A fairness index value close to 1 indicates an equal repartition of the load, while a totally unfair situation is reflected by an index value of $1/n$, where n is the number of nodes. The energy consumption is not very well distributed when the number of brokers is low and fairness increases with the number of brokers. Among the different metrics, the less efficient strategies in terms of average energy consumption lead to the best fairness results. This tends to indicate that the most efficient strategies only alleviate the load of a reduced set of nodes. Using a dominating set leads a good repartition of the energy consumption.

Figures 2(a) and 2(b) represent the same evaluations when the network is organized as a tree by RPL. The general shape of the energy consumption profiles remain similar, even

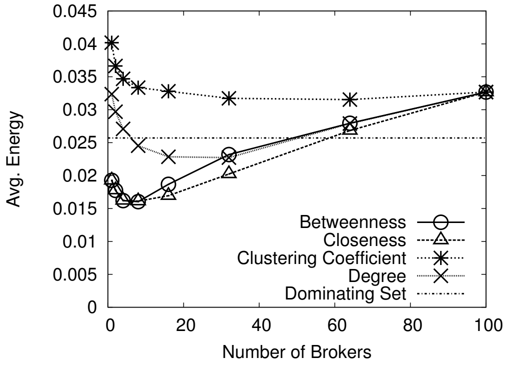


(a) Average Energy (W.h)

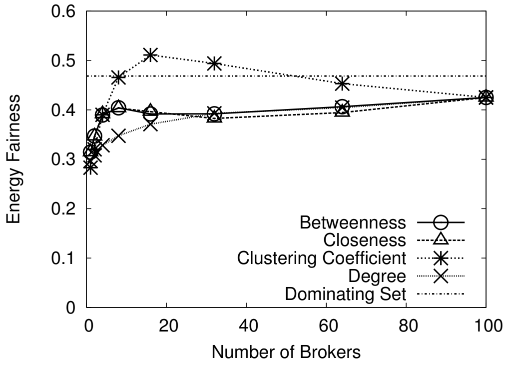


(b) Energy Fairness

Fig. 1. Energy consumption profile for 90 publishers and 20 subscribers using shortest paths routing



(a) Average Energy (W.h)



(b) Energy Fairness

Fig. 2. Energy consumption profile for 90 publishers and 20 subscribers using RPL-like routing

though the dominating set approach has a lower performance. Degree, on the other hand, results in a good performance for 20 brokers. We can also note that the average energy consumption increases globally of about 50 % when compared to the shortest paths case, which fits the increase in routes lengths, as shown on Figures 3(a) and 3(b).

These two figures show the evolution of the routes length between publishers and subscribers when the number of brokers increase in both routing situations. We may notice that route length is not correlated to average energy spent or to fairness, as the route length constantly decreases when the number of brokers increases, while energy consumption presents a minimum.

Energy consumption fairness, represented on figure 2(b) is much worse with RPL than in the shortest paths case, as all the traffic passes by nodes that are close to the tree root. The relative performance of the different criteria is not maintained either. The dominating set approach still has a good performance. However, when the number of brokers is reasonable, all metrics lead to better results, except the clustering coefficient.

Figures 4(a) and 4(b) represent the network maximum sustainable load. Combining these figures with the energy consumption ones, we can see that there is an optimal operation region. When shortest paths are used, the dominating approach gives the best performance in every situation, but when RPL is used, closeness and betweenness centralities give better results for the same number of brokers as the one that minimizes energy. This point shall be the target of an operational algorithm.

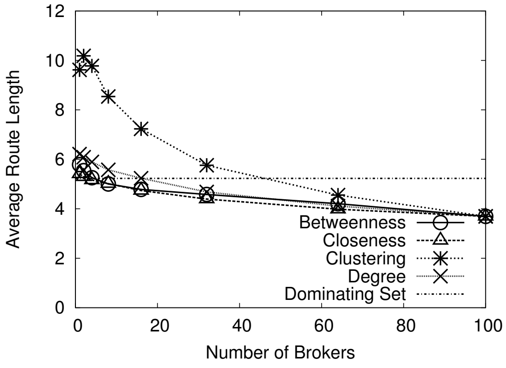
Finally, figures 5(a) and 5(b) represent, with RPL, the average delivery time and queues length that remain relatively stable for most metrics, except for the clustering coefficient. This indicates that the maximum load identified above can actually be applied in the network.

C. Decreasing the number of subscribers

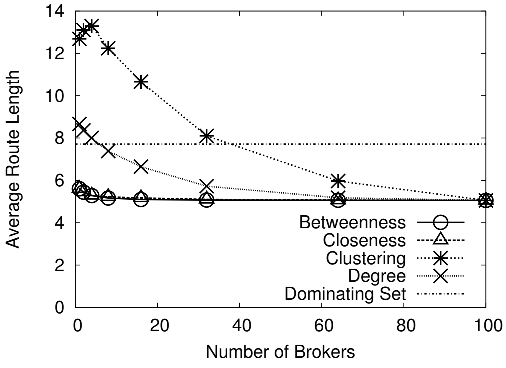
Figures 6(a), 6(b) and 6(c) represent, respectively, the performance metrics when there is only a single subscriber for 90 publishers and when RPL is used. These graphs show that in this case, the best strategy from the energetic point of view, which is also the less complex, consists in selecting every node as to act as a broker. In this case average energy is minimal and the fairness is reasonable. However, in this case the achievable load decreases when the number of brokers increases, as data replication level increases with the number of brokers.

D. Increasing the number of subscribers

Figures 7(a), 7(b) and 7(c) represent, respectively, the performance metrics when there is only a single publisher for 90 subscribers and when RPL is used. These graphs are extremely

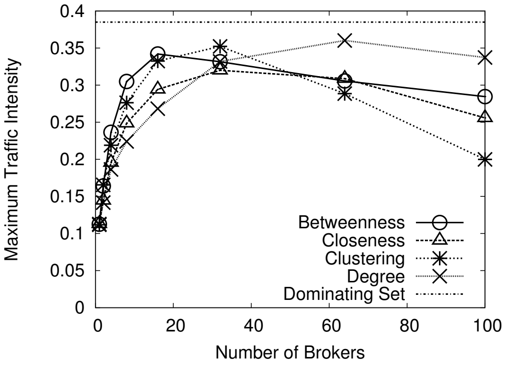


(a) True Shortest Paths

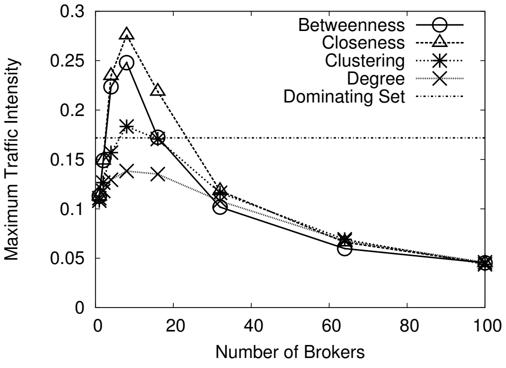


(b) RPL-like tree

Fig. 3. Routes lengths for 90 publishers and 20 subscribers with and without RPL



(a) True Shortest Paths



(b) RPL-like tree

Fig. 4. Maximum Sustainable Load for 90 publishers and 20 subscribers with and without RPL

similar to the ones obtained in our base scenario (90 publishers and 20 subscribers) but shifted. Achievable load is multiplied by a factor of 20 while energy consumption is reduced by a similar factor. Fairness is halved.This indicates that it is the number of subscribers that, in the end, defines how the performance of the network evolves and that the number of publishers influences the absolute value of the figures.

E. Other scenarios

We also evaluated the situations in which the number of publishers and subscribers were similar in the network From all these simulations, we can see that as the number of subscribers increases, most of the centrality measures present an optimal functioning point that is around 16 to 32 brokers for a 100 nodes network . However, when the number of subscribers gets large, the dominating set approaches leads to the best average energy in the shortest paths case, regardless of the number of brokers. When RPL is used, selecting brokers according to the nodes closeness and betweenness centrality values still manage to achieve a better performance. Similar conclusions hold for the maximum sustainable load.

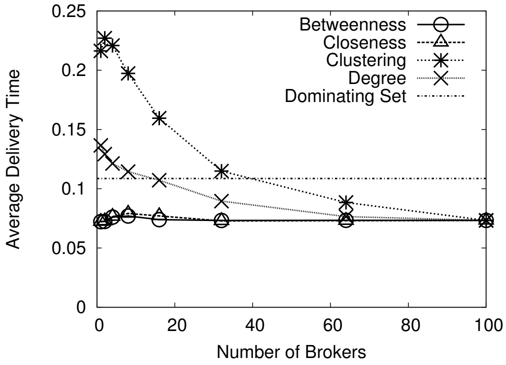
Increasing the network density or diameter by playing with the transmission range does not change the results beyond what can be expected from the modification in route lengths and traffic intensities.

VI. CONCLUSION AND FUTURE WORKS

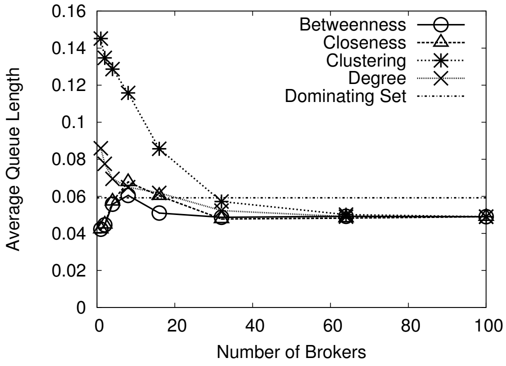
In this paper, we compared various approaches to select the brokers in a multihop wireless sensor network. Using a model of energy consumption that takes into account overhearing and a queueing model to evaluate the load of each node, we were able to compare the performance of various strategies to select a set of brokers among the candidate nodes.

From the simulation results we obtained, we can draw a few conclusions. In a fair number of scenarios, building a dominating set gives a good performance, even though it may limit the gain especially when RPL or a similar tree-building algorithm is used. However, RPL and tree-like structures are often more than a design choice in sensor networks. They may be necessary as soon as an on-demand routing protocol would become too costly. In all these situations, betweenness and closeness centrality measures clearly have the best optimization potential.

The simulations results we presented here shall be complemented with further analysis of the duty cycling procedure for example, which will reduce the part of energy consumption due to overhearing. Control traffic shall also be evaluated, as increasing the number of brokers means that the volume of such traffic for synchronization and tables exchange will increase. Fault tolerance is also a key issue, as the brokers will eventually fail when their battery level reaches zero.



(a) Average delivery time



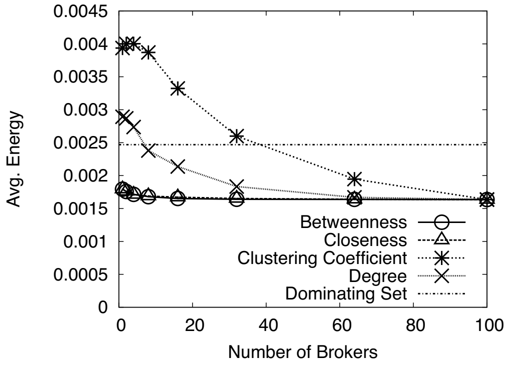
(b) Average queue load

Fig. 5. Queues load and delivery times for 90 publishers and 20 subscribers with RPL

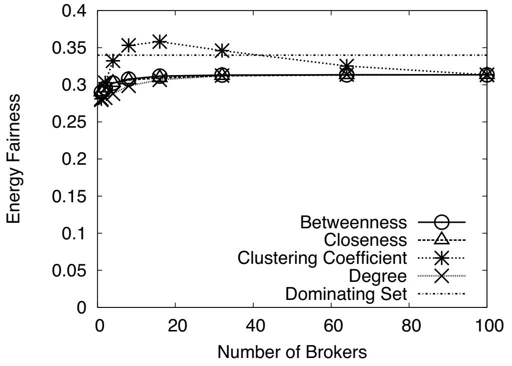
Remaining energy should therefore weight the centrality-based metric, which shall be computed using only local information. As [22] and [23] showed that an egocentric (i.e. localized) computation of the betweenness centrality gave results that were close to the global computation in many scenarios, and as we are only interested in the ranking that this centrality measure provides, the egocentric betweenness centrality is a promising criterion.

Once every node is able to calculate its own score, we can modify a distributed clustering algorithm such as [27] that relies on the comparison of a metric to elect clusterheads. Such algorithms are self-stabilizing, which is a desirable property in our case. Self stabilization is achieved because there is a procedure of de-election of a node based on topological criteria. When two (or k) nodes get close to each other, they cannot all remain clusterheads. We need to derive an equivalent criterion to allow nodes to leave the brokers set before an equivalent algorithm is possible. However the design of this stopping criterion is far from being trivial, as it should target the correct number of brokers that brings the network in its optimal operation zone, which may depend on the scenario.

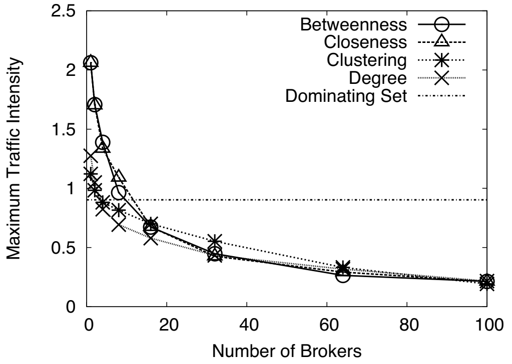
Once such a distributed algorithm is built, the subsequent step will be to translate it into a protocol and to evaluate it in a network simulator and on an experimental platform. These steps will allow us to evaluate the effect of radio channel and to compare energy consumption figures with real hardware.



(a) Average Energy (W.s)



(b) Energy Fairness



(c) Maximum Load

Fig. 6. Base performance metrics for 90 publishers and a single subscriber using RPL

ACKNOWLEDGMENTS

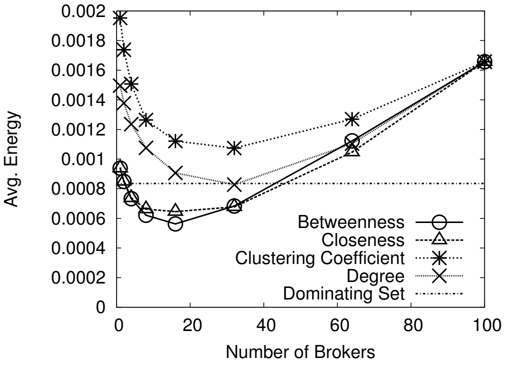
This work is funded by the Diaforus project, granted by the French National Research Agency (ANR).

REFERENCES

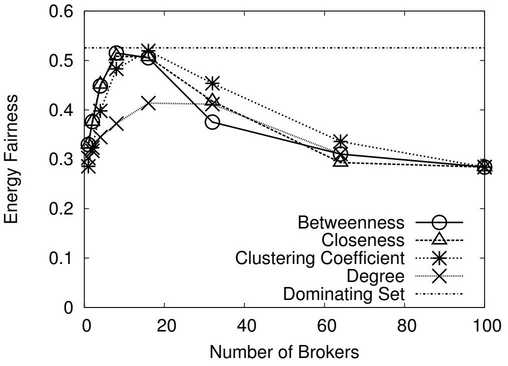
[1] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, "Design and evaluation of a wide-area event notification service," *ACM Transactions on Computer Systems*, vol. 19, no. 3, pp. 332–383, Aug. 2001.

[2] G. Cugola, E. Di Nitto, and A. Fuggetta, "The JEDI event-based infrastructure and its application to the development of the OPSS WFMS," *IEEE Transactions on Software Engineering*, vol. 27, no. 9, pp. 827–850, Sep. 2001.

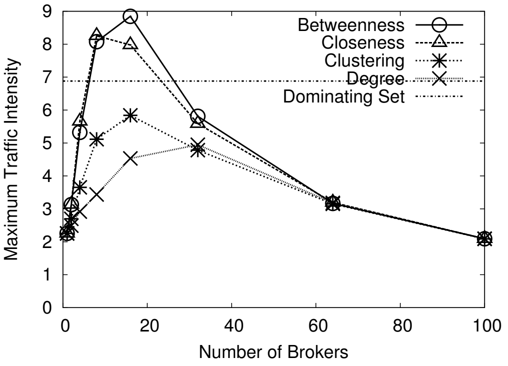
[3] W. W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A. P. Buchmann, "A peer-to-peer approach to content-based publish/subscribe," in *DEBS '03*, 2003.



(a) Average Energy (W.s)



(b) Energy Fairness



(c) Maximum Load

Fig. 7. Base performance metrics for 90 publishers and a single subscriber using RPL

[4] F. Cao and J. P. Singh, "Efficient event routing in content-based publish-subscribe service networks," in *IEEE INFOCOM 2004*, Hong Kong, China, Mar. 2004.

[5] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *Journal of Selected Areas in Communications*, vol. 20, no. 8, 2002.

[6] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz, "Bayeux: an architecture for scalable and fault-tolerant wide-area data dissemination," in *NOSSDAV '01*, 2001.

[7] A. Gupta, O. D. Sahin, A. Divyakant, and A. El Abbadi, "Meghdoot: Content-Based Publish/Subscribe over P2P Networks," in *MIDDLEWARE 2004*, 2004.

[8] E. Di Nitto, D. J. Dubois, and R. Mirandola, "Overlay self-organization for traffic reduction in multi-broker publish-subscribe systems," in *ICAC '09*. Barcelona, Spain: ACM, 2009.

[9] A. R. Khakpour and I. Demeure, "Chapar: A Cross-Layer Overlay Event System for MANETs," in *MobileWireless Middleware, Operating Systems, and Applications*. Springer Berlin Heidelberg, 2009, vol. 7, pp. 325–339.

[10] D. A. Tran and L. H. Truong, *Advances in Next Generation Services and Service Architectures*. River Publishers, 2011, ch. Chapter 36 - Enabling Publish/Subscribe Services in Sensor Networks.

[11] U. Hunkeler, H.-L. Truong, and A. Stanford-Clark, "MQTT-S: A publish/subscribe protocol for wireless sensor networks," in *Workshop on Information Assurance for Middleware Communications (IAMCOM '08)*, Bangalore, India, Jan. 2008.

[12] S. Rooney and L. Garcés-Erice, "Messo & preso practical sensor-network messaging protocols," in *ECUMN*, Toulouse, France, Feb. 2007, pp. 364–376.

[13] S. Pleisch and K. Birman, "SENSTRAC: Scalable Querying of SENSor Networks from Mobile Platforms Using TRACking-Style Queries," *International Journal of Sensor Networks*, vol. 3, no. 4, 2008.

[14] E. G. Davis and A. C. Augé, "Presence-based architecture for wireless sensor networks using publish/subscribe paradigm," in *9th IFIP TC 6 international conference on Wired/wireless internet communications (WWIC'11)*, 2011.

[15] Y. Liu, B.-C. Seet, and A. Al-Anbuky, "Virtual brokers for large-scale publish/subscribe in wireless sensor networks," in *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '10)*, Hong Kong, China, Dec. 2010.

[16] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Journal of Computer Communications, Special Issue on Network Coverage and Routing Schemes for Wireless Sensor Networks*, vol. 30, no. 14-15, Oct. 2007.

[17] C. Chaudet, I. Demeure, and S. Ktari, "A model to evaluate brokers overlays for publish/subscribe in wireless sensor networks," in *International Conference on Wireless On-demand Network Systems and Services (WONS 2012)*, Courmayeur, Italy, Jan. 2012.

[18] IEEE Standards, "IEEE 802.15.4: Wireless Medium AccessControl (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," Oct. 2003.

[19] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," Internet Draft – draft-ietf-roll-rpl-19, Mar. 2011.

[20] L. Jia, R. Rajaraman, and R. Suel, "An efficient distributed algorithm for constructing small dominating sets," in *20th ACM Symposium on Principles of Distributed Computing (PODC)*, 2001.

[21] L. C. Freeman, "Centrality in social networks: Conceptual clarification." *Social Networks*, vol. 1, no. 3, 1979.

[22] P. V. Marsden, "Egocentric and sociocentric measures of network centrality," *Social Networks*, vol. 24, 2002.

[23] M. Everett and S. P. Borgatti, "Ego network betweenness," *Social Networks*, vol. 27, 2005.

[24] A. Cuzzocrea, A. Papadimitriou, D. Katsaros, and Y. Manolopoulos, "Edge betweenness centrality: A novel algorithm for qos-based topology control over wireless sensor networks," *Journal of Network and Computer Applications*, Jul. 2011.

[25] E. M. Oliveira, H. S. Ramos, and A. A. Loureiro, "Centrality-based routing for wireless sensor networks," in *IFIP Wireless Days (WD)*, Oct. 2010.

[26] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report TR-301, Tech. Rep., 1984.

[27] C. Johnen and L. H. Nguyen, "Robust self-stabilizing clustering algorithm," in *10th International Conference On Principles Of Distributed Systems (OPODIS 2006)*, Bordeaux, France, Dec. 2006.