ICST Transactions on Ubiquitous Environments Research Article ICST.ORG

Towards energy-aware semantic publish/subscribe for wireless embedded systems

Davy Preuveneers[1,*], Yolande Berbers[1]

[1]IBBT-DistriNet, Department of Computer Science, KU Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium

Abstract

In this article, we present our lightweight semantic publish/subscribe system μC-SemPS that is targeted towards wireless battery-powered embedded systems in ubiquitous computing environments. The key challenge that we address is the minimization of the overall energy consumption, which involves computational aspects for semantic subscription matching, and communication aspects for the delivery of the publications as events in a wireless network. Our pub/sub system relies on a compact mathematical representation of the semantic subscriptions and publications, as well as fast semantic matching and routing algorithms. From an energy efficiency point of view, it favors computation over the more expensive wireless communication for the semantic matching and routing of events. When compared to more conventional pub/sub routing implementations, experimental results from network and energy simulations with MiXiM, an OMNeT++ modeling framework extension for wireless and mobile networks, show that our approach for routing semantic events significantly reduces energy consumption.

Received on 18 August 2011; accepted on 2 July 2012; published on 14 November 2012

Keywords: energy-awareness, publish/subscribe, semantic routing, wireless embedded systems

doi:10.4108/trans.ubienv.2012.10-12.e1

1. Introduction

The publish/subscribe (pub/sub) communication paradigm [5] provides a many-to-many anonymous event communication model that is decoupled in time and space, and as such an ideal distributed notification platform for ubiquitous computing systems. Traditionally, an event is a collection of attributes as key-value pairs, structured according to an a-priori known event schema, and communicated by means of push/pull interactions between possibly mobile publishers and subscribers. An event notification service acts as the mediator to notify subscribers about published events that match the constraints in at least one of their subscriptions. For scalability reasons, the event notification service is often implemented as an overlay network of event brokers [9] that match and route events/subscriptions to other brokers and clients.

*Corresponding author. Email: davy.preuveneers@cs.kuleuven.be

The major challenge that we address in this article is the energy-efficiency of semantic pub/sub systems for wireless and battery-powered embedded systems. This class of pub/sub systems [2, 3, 11, 12] is gaining importance given the fact that (1) emerging computing paradigms like Machine-to-Machine (M2M) and the Internet of Things (IoT) will generate ever-increasing amounts of data, and that (2) participants in loosely-coupled heterogeneous systems will use a variety of terminologies. The fact that all events must be unambiguously interpreted by the publishers and subscribers is the main motivation why semantics must become a first class entity of the pub/sub system.

In this article, we investigate how a pub/sub system can be optimized for battery-powered wireless sensor nodes and embedded microcontrollers (μC), by matching and routing semantic events in such a way that reduces computation and communication energy usage. Addressing both challenges at the same time is not straightforward. As previous research [4] has shown, communication can be much more expensive

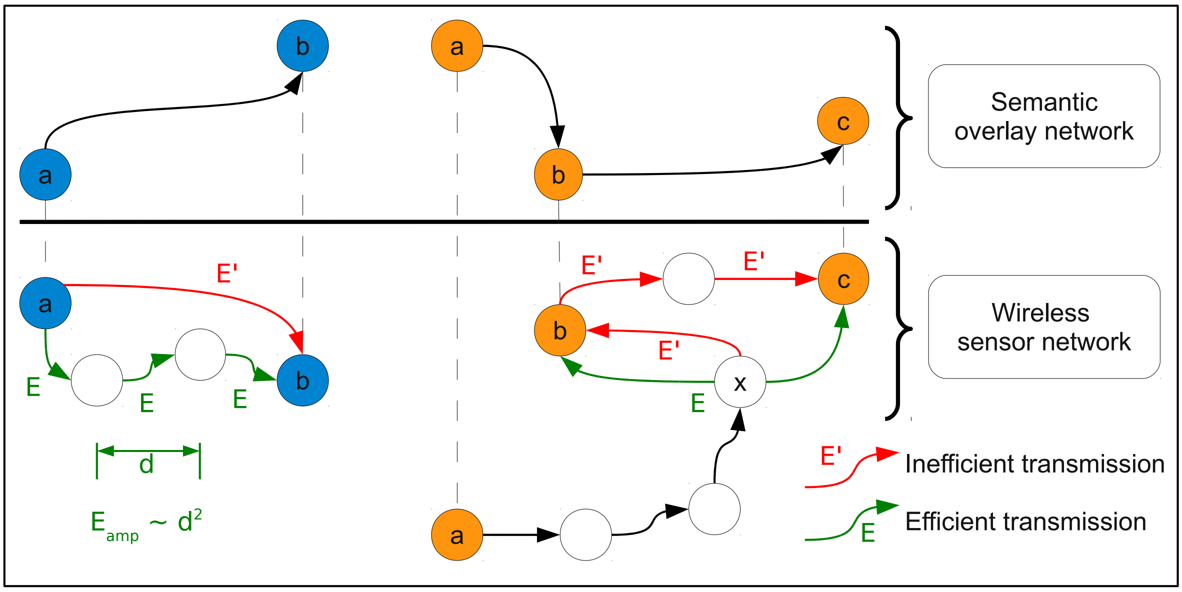

Figure 1. Mapping of a semantic overlay structure onto a peer-to-peer network

than computation, and therefore power-efficient network protocols [6] are often part of the big picture as far as minimizing energy consumption in wireless sensor networks is concerned. However, merely implementing a semantic pub/sub system as an overlay structure on top of a wireless peer-to-peer network and relying on an power efficient communication protocol – hence treating both problems as separate concerns – can still be inefficient from an energy perspective. A semantic overlay network typically imposes a network structure with routing paths from event producers to their respective subscribers where the order of the subscribers is defined by the specificity of their subscriptions. When routing events from nodes with generic subscriptions to nodes with more specific subscriptions (i.e. imposing a subset of matching events), the forwarding of events stops at the node for which the event no longer matches with its subscriptions. The semantic overlay network provides a means to route events while avoiding passing through nodes without any matching subscriptions. However, strictly following the routing path imposed by the overlay network may not be efficient from an energy point of view depending on the distance between the nodes involved. This is illustrated in Figure 1 and explained further below.

A direct unicast between two nodes may theoretically be more expensive than sending over intermediate nodes with minimum transmission energy (MTE) as routing criteria, because the energy consumption of the wireless signal amplifier is quadratically proportional to the distance. On the left of Figure 1, a message is transmitted with direct unicast from a to b requiring E' nJ/bit, or through $m-1$ intermediate nodes requiring $m.E$ nJ/bit. For transmission over larger distances, the latter is often more efficient (depending on the number of intermediate nodes, the distance between them, and the energy dissipated in the transmitter/receiver/amplifier electronics). However, the decision is not clear cut. If the batteries of the intermediate nodes are almost depleted, a direct unicast can improve the network lifetime. See [6] for an analysis of when direct unicast is cheaper than MTE. As such, there is a clear trade-off between being more energy efficient and reducing the overhead of routing through nodes without matching subscriptions.

The problem is worsened for multi-hop communication in the semantic overlay network, if each hop in the semantic overlay network is mapped onto direct unicast or MTE in the physical network. On the right in Figure 1, we see how a direct mapping of the routing in the semantic overlay network can lead to inefficient routing paths in the physical network. The event routing from b to c can be avoided by a multicast/broadcast at node x to reach nodes b and c.

In this article, we present μC-SemPS, a semantic event pub/sub system for battery-powered micro-controllers in mobile and ubiquitous computing environments that aims to optimize the network lifetime. The main contributions are:

- An event subscription model for efficient semantic pub/sub matching
- A semantic-based clustering approach for efficient event routing

- A significant energy usage reduction vs. more conventional pub/sub schemes

The article is structured as follows. After discussing related work in section 2, we present in section 3 our semantic subscription model. The semantic pub/sub middleware and algorithms are explained in section 4. We elaborate on the MiXiM-based experimental results in section 5. A concluding overview of our contributions and opportunities for further work are highlighted in section 6.

2. Related work

The publish/subscribe paradigm has been extensively studied as a way for selective information dissemination. Most of the work in this area deals with the efficient dissemination of simple subscription formats in subject-based and content-based systems. Recently, semantic matching in pub/sub systems is receiving more and more attention to achieve higher event expressiveness.

Current solutions attempt to build semantic-awareness with a simple hierarchical topology of concepts (taxonomies of classes and properties). S-ToPSS [12] extends the conventional key-value pair-based systems with methods to process syntactically different, but semantically equivalent information. S-ToPSS uses an ontology with synonyms, a taxonomy and transformation rules to deal with syntactically disparate subscriptions and publications.

The Ontology-based Publish/Subscribe system (OPS) [18] represents subscriptions as RDF graph patterns and uses a subgraph isomorphism algorithm for efficient matching. Follow-up research on semantic pub/sub systems of some of the authors resulted in the Semantic Publish/Subscribe System (SPS) [11], which leverages more of the expressive power of OWL Lite but relies on a centralized broker with a powerful server to do the semantic matching.

Chand *et al.* present interest similarity metrics [2] for peer-to-peer overlays of pub/sub networks to tackle the lack of expressiveness in many subscription languages. The containment hierarchy tree that is implied by the containment-based and similarity-based proximity metrics is similar to class and property subsumption relationships in ontologies. Subsumption is a description logics concept which also defines a partial order of containment on the individuals. Other metrics like those proposed in [21] use fuzzy semantic matching to eliminate the need for exact similarities, but go beyond the capabilities of embedded devices.

Many of the above semantic pub/sub systems go beyond the capabilities of embedded devices. The DSWare [10] middleware offers publish/subscribe services to distributed wireless sensor networks and supports the specification and detection of patterns of compound events.

Resource-aware publish/subscribe in wireless sensor networks is studied in [17]. The authors propose a protocol that aims to extend the network's lifetime by offering tradeoffs between fixed event dissemination paths that increase communication efficiency, and resource-awareness that provides freedom for event routing. Network hops are used as a measure for the delivery cost. With energy efficiency being the main concern, hops are not a good measure for quantifying energy usage. Energy consumption of a wireless amplifier increases quadratically with the hop distance, so that transmitting over smaller distances (i.e. with more hops) could be more optimal.

In [8], Demeure *et al.* present an energy-aware middleware for MANETs of handheld devices which includes a publish/subscribe event system. The middleware is made energy-aware by providing policies that specify adaptations in the middleware for a variety of battery thresholds. These adaptations may trigger the use of more energy-friendly encryption algorithms or change communication by using a non-acknowledged protocol to reduce network activity. The authors do not immediately focus on optimizing the event matching and routing. For an overview of energy-aware routing in wireless sensor networks, we refer to the survey of Akkaya *et al.* [1].

3. Semantic event subscription model

Before we discuss the event and concept models used in μC-SemPS, our semantic pub/sub system, we will briefly describe the energy model that we use to simulate energy consumption due to computation and communication. The energy model is fairly simple and assumes ideal operational circumstances, but helps us to simulate large scale sensor networks and compare different algorithms under the same working conditions.

3.1. Energy usage approximation model

We reuse the same first order radio model for communication presented in the widely referenced work [6] by Heinzelman *et al.* This is a simple model where the radio dissipates $E_{elec} = 50nJ/bit$ to run the transmitter and receiver circuitry and $\epsilon_{amp} = 100pJ/bit/m^2$ for the transmit amplifier. To transmit a k-bit message a distance d away, the radio dissipates:

$$E_{Tx}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2$$

To receive this message, the radio dissipates:

$$E_{Rx}(k) = E_{elec} * k$$

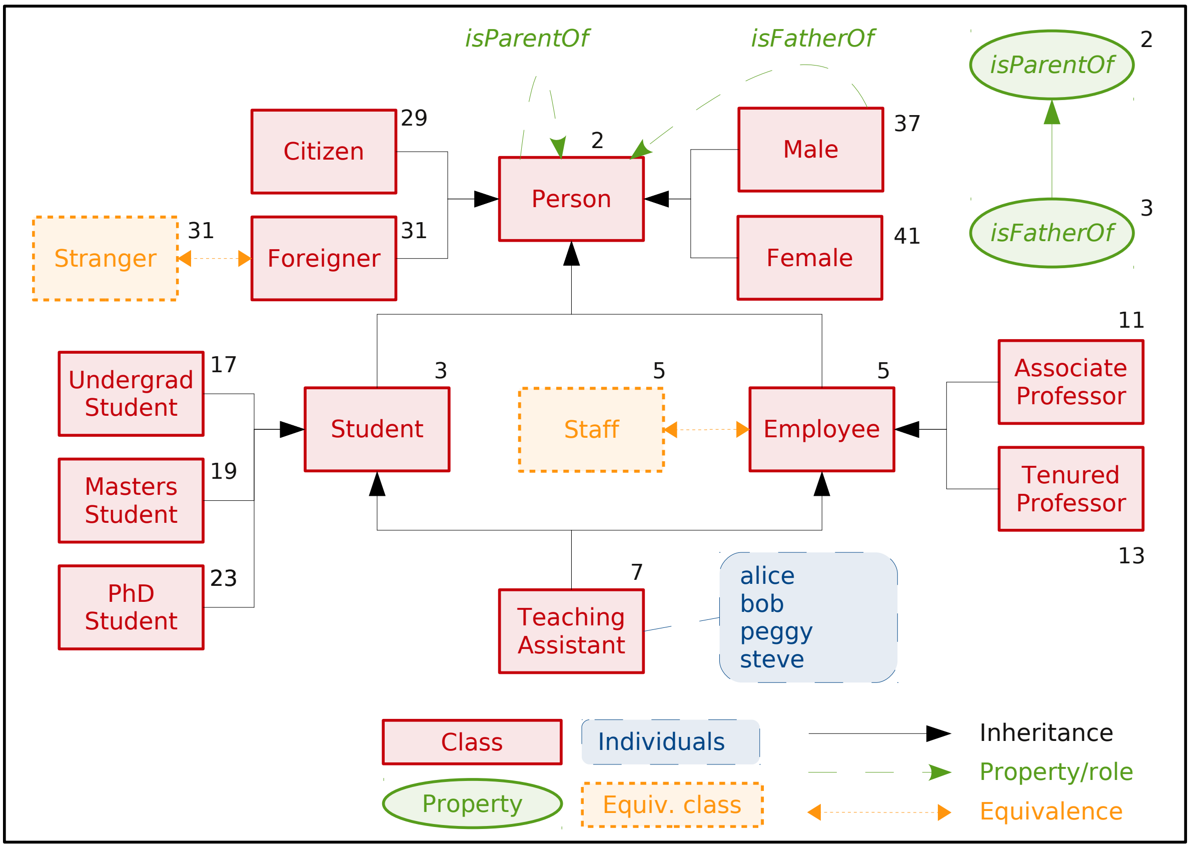

Figure 2. Inheritance of ontology classes and properties with prime number assignment

For a publisher to send a message of 64 bytes (or 512 bits) with direct unicast to a subscriber 50 meters away, the total energy consumption for communication would be:

$$
\begin{aligned}
E_{direct} &= E_{Tx}(512, 50) + E_{Rx}(512) \\
&= 2 * 50nJ/bit * 512bits \\
&+ 100pJ/bit/m^2 * 512bits * 2500m^2 \\
&= 132\mu J
\end{aligned}
$$

Routing the message over the same distance but with retransmissions by four intermediate nodes at 10 meters apart would require:

$$
\begin{aligned}
E_{mte} &= 5 * \{E_{Tx}(512, 10) + E_{Rx}(512)\} \\
&= 5 * \{2 * 50nJ/bit * 512bits \\
&+ 100pJ/bit/m^2 * 512bits * 100m^2\} \\
&= 46\mu J
\end{aligned}
$$

For the energy dissipated due to computation, measurements on a variety of microprocessors have shown that the energy usage varies between 0.5∼4 nJ/instruction. Shnayder *et al.* measured power consumption of the Mica2 mote in [15] with similar characteristics. For our simulated experiments, we will use an estimate of 2 nJ/instruction which matches pretty well with what some ARM processors consume.

Note that this energy usage approximation model makes certain assumptions that may be hard to achieve in a particular implementation or setting. For example, we assume ideal circumstances for communication where sending and receiving nodes are awake during transmission so that messages do not require any retransmissions between these nodes. This will be a concern when nodes are not always on at the same time. When a receiving node goes to sleep to save energy, then a sending node may have to retransmit to get the message across. The impact of timing and coordination issues will be more outspoken for transmitting messages between wireless nodes over longer distances and/or multiple hops.

3.2. Concept model

Concept-based semantic publish/subscribe systems describe events and subscriptions at a higher level of abstraction than simple key-value pairs. The semantic relationships are defined in ontologies for an unambiguous interpretation of the event structure. In Figure 2, we illustrate this with a familiar example often used for educational purposes. The concept model to define the semantic meaning of a subscription or event consists of the following constructs:

- **Inheritance hierarchy of named classes:** A class (or concept) can have multiple parent classes and subclasses. Any individual (entity or instance of a class) of a class is also a member of its ancestor classes:

$$Student \subseteq Person$$

- **Inheritance hierarchy of properties:** An (object or datatype) property defines a role of a class by pointing to another class. A similar inheritance relationship can also be defined for properties (or roles):

$$isFatherOf \subseteq isParentOf$$

- **Equivalence of classes and properties:** Classes and properties are defined semantically equivalent if they represent the same set of individuals. For example, the class *Car* could be a synonym for the class *Automobile*:

$$Car \equiv Automobile$$

  Compared to other semantic pub/sub systems, our model also supports equivalence between properties and classes. For example, the class $Aircraft$ is equivalent to the class $Vehicle$ for which a property $hasWings$ is defined (We assume that $Aircraft \subseteq Vehicle$ and that $hasWings$ has other domains like the class $Animal$).

$$Aircraft \equiv (Vehicle \wedge hasWings)$$

- **Anonymous classes:** These are classes for which no name is specified. They define a set of individuals that comply with a set of constraints. Examples include the union or intersection of classes, property restrictions on domain or range, complements of classes, etc.

- **Custom mappings:** These are equivalence relationships defined on the individual level rather than on the class level. They allow for expressing types of semantic similarity that cannot be expressed in the class or property hierarchies.

Synonyms and the two types of inheritance relationships are typically found in most semantic pub/sub systems. The acyclic, reflexive and transitive inheritance relationship is in the literature sometimes referred to as *containment*, *type inclusion* or *subsumption*.

3.3. Event model and subscription language

The event model specifies the organization of data inside the events and provides an expressive subscription language for subscribers to define their interests in certain events. Rather than using a directed labeled graph representation for the semantic relationships of an event of interest, we use the user-friendly compact Manchester syntax for OWL [7] to express semantic subscriptions. For example,

Student **or** Person **that** hasAge **all** xsd:integer[<= 25]

would represent with the usual precedence rules an interest in students and persons that are 25 years or younger. Rather than relying on subgraph isomorphisms, we make use of an efficient encoding of semantic relationships based on the properties of prime numbers, as explained in the following subsection.

3.4. Semantic event matching

The encoding presented in this article is proposed to counter one of the major disadvantages of ontologies: the reasoning tools and technologies underlying the semantic modeling with ontologies are particularly expensive in terms of memory and processing time. They are not designed with the limited computational resources of an embedded system environment in mind. The prime number based encoding and matching algorithms we presented in [14] allow for embedding semantic awareness in resource constrained devices. The algorithms were developed with the aim to reduce the memory and processing complexity with several orders of magnitude compared to other standard ontology reasoning engines. The compact representation of the semantic relationships presented in this paper is also useful from a communication perspective because the message lengths of the events are significantly smaller. We will briefly summarize some terminology before highlighting the basic principles of our event matching algorithms:

Hierarchy: This is the partial ordered set of classes (or properties) that reflects the subsumption relationships and is represented with a symbol χ.

Child: A class $A \in \chi$ is a child of a class $B \in \chi$ if and only if class A has a *direct* connection with class B and inherits from class B in the hierarchy. A *Leaf* class is a special child in the hierarchy that has no children of its own. The child relationship is denoted as: $A <:_d B$

Descendant: A class $A \in \chi$ is a descendant of class $B \in \chi$ if and only if it is either a child of class B or a descendant of one of the children of class B. This recursive definition of descendant is denoted as: $A <: B$. A child is a *direct* descendant, hence the subscript d in the child notation.

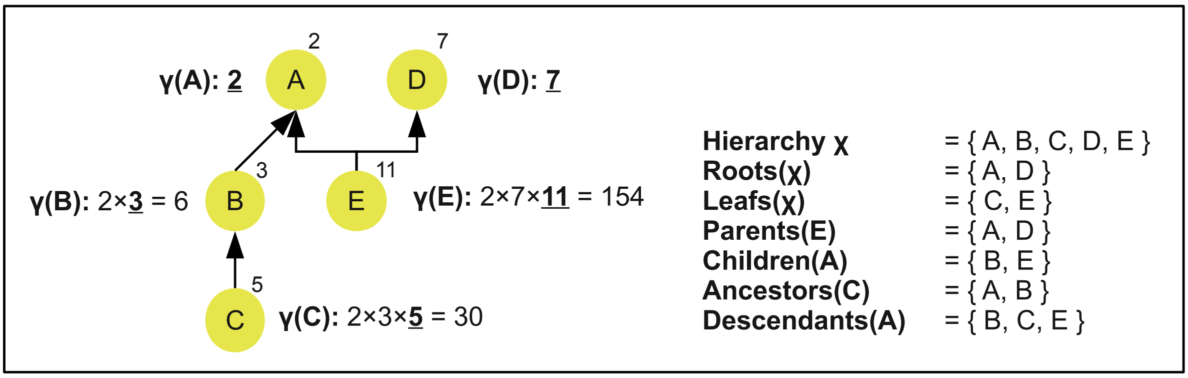

Figure 3. Hierarchy encoding using prime numbers

Equivalence: Two classes A and B in the hierarchy χ are defined equivalent if both classes have the same set of instances or individuals at all times. The equivalence relationship is denoted as: $A \equiv B$

Subsumption: Refers to the reflexive, transitive and anti-symmetric relationship of classes in the hierarchy χ, which states that a class $A \in \chi$ is subsumed by a class $B \in \chi$ if and only if the set of instances of class A are also included in those of class B. This is the same as stating that class B subsumes class A. The subsumption relationship is denoted as: $\gamma(A) \subseteq \gamma(B)$. Note that: $\gamma(A) \subseteq \gamma(B) \Leftrightarrow A \equiv B \vee A <: B$.

Gene: A gene is a unique symbolic or numeric identifier that is used to define a subsumption relationship between two classes. If a class A inherits from a class B, then it inherits all the genes of class B. Each class has a new differentiating gene that it only shares with its descendants. The differentiating gene of class A is represented as g_A.

Code: A class is encoded into a compact representation that allows efficient subsumption testing. For bit vector representations, the encoding is often determined by first assigning to each gene a distinguishing bit position in the bit vector, and then for each class setting the corresponding bits of its genes to 1. The code of a class A is represented as $\gamma(A)$.

Given these definitions and notations for specifying relationships between classes in a multiple inheritance hierarchy, we declare the following functions for a class $A \in \chi$:

$$\begin{aligned}
parents(A) &= \{B \in \chi | A <:_d B\} \\
children(A) &= \{B \in \chi | B <:_d A\} \\
ancestors(A) &= \{B \in \chi | A <: B\} \\
descendants(A) &= \{B \in \chi | B <: A\} \\
roots(A) &= \{B \in ancestors(A) | parents(B) = \emptyset\} \\
leaves(A) &= \{B \in descendants(A) | children(B) = \emptyset\}
\end{aligned}$$

Semantic event matching requires an efficient encoding for subsumption testing. Bitvectors do not work well for ontologies because publishers and subscribers do not know all the concepts in advance (closed world assumption). Also, representing inheritance relationships as bitvectors results in a sparse binary matrix, i.e. the matrix contains many 0's, because the size of the hierarchy is usually much larger than the branching factor of a class in the hierarchy. Therefore, we will only encode a reference to the ancestors in the representation of a class by assigning a number as differentiating gene $g \in G$ to each class that is co-prime with all the other genes. The encoding $\gamma(A)$ of class A can then be defined as the multiplication of its differentiating gene g_A with the inherited genes of its ancestors. This way, a class $A \in \chi$ subsumes a class $B \in \chi$ if and only if the gene $g_A = \varphi(A)$ of class A divides the encoding $\gamma(B)$ of class B.

The encoding of a multiple inheritance hierarchy $\chi = \{A, B, C, D, E\}$ is illustrated in Figure 3. The differentiating gene of each class is underlined. For example, class E inherits the genes 2 and 7 from its ancestors and is assigned 11 as its differentiating gene g_E. The encoding of class E becomes $\gamma(E) = 2 \times 7 \times 11 = 154$. As classes A and D have no ancestors, their encoding $\gamma(A)$ and $\gamma(D)$ is solely based on their differentiating gene $g_A = 2$ and $g_D = 7$. The degree of compaction solely depends on the prime number p that is assigned to a class C. The objective is to assign prime numbers to classes in such a way that after multiplication the length of the encoding – either for the individual class with the longest encoding or the total encoding length for all classes

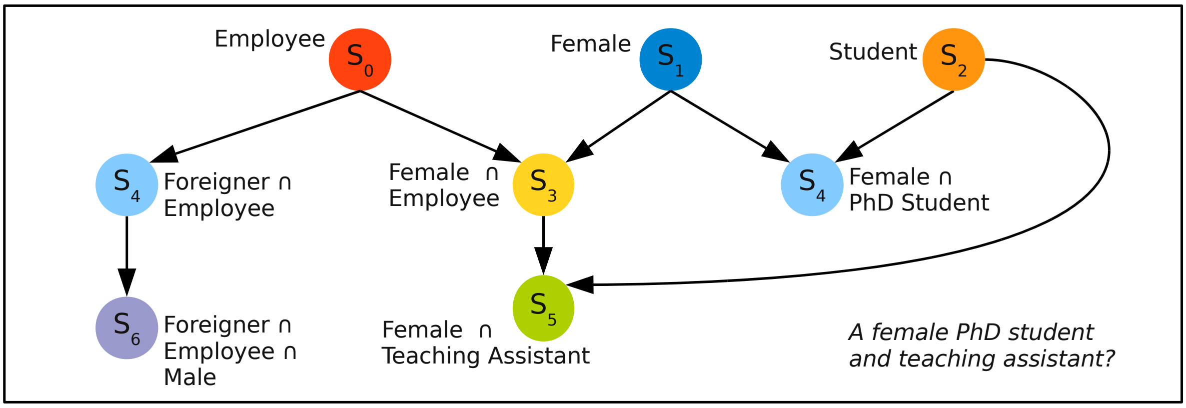

Figure 4. Organizing nodes according to the specificity of their subscriptions

together – is minimized. See [13, 14] for more details on the heuristics we used to optimize the encoding length of each class. For the very simple hierarchy in Fig. 3, the OWL ontology file is more than 1500 bytes long, whereas our encoding generates an equivalent representation of less than 100 bytes:

```
A=2,2
B=3,6
C=5,30
D=7,7
E=11,154
```

The advantages of our semantic encoding are:

- A class (or property) can be identified by a simple prime number rather than a possibly long string that represents its URI.
- All the relevant semantic relationships are embedded in the encoding of each class and property.
- Partial imports of an ontology are possible because you only need the class and property encodings of the terminology you are interested in.

These characteristics have an immediate effect on the message length of events that have to be routed, as textual descriptions are being replaced with a simple prime number. Message size reductions of a factor of 50 or more were not uncommon in our experiments. Also, because all the relationships are already explicitly represented in the class encoding, we can eliminate the computational overhead of classifying an ontology (a logical inference process that makes all implicit semantic relationships explicit) at runtime. The fact that our representation is line oriented makes it also easier to parse than XML-based file formats.

The above improvements also help to reduce the size of the subscriptions. For example, a subscription for tenured professors who are the grandfather of a student, would be in the compact OWL Manchester syntax:

TenuredProfessor **that**
isFatherOf **some** (Person **that** isParentOf **some** Student)

This is twice an intersection of a class and property restriction. Each label is replaced with the corresponding prime number (see Fig. 2 for the prime number assignment):

$$13 \cap \exists\ 3\ (2 \cap \exists\ 2\ 3)$$

Any individual with a class encoding that can be divided by 3 is a *Student* (or a subclass thereof). For example, a *PhD Student* has encoding $2 * 3 * 23 = 138$ and can be verified to be an instance of both *Person* and *Student*. The same reasoning can be applied with 2 and 13 for the classes *Person* and *Tenured Professor* respectively, as well as for properties. Note that in many cases we do not have to carry out the division, but use faster heuristics to rule out a semantic match. For example, to optimize the encoding length we ensure that the differentiating gene of a subclass is always bigger than that of its parent. Therefore, we can quickly rule out that *C* does not inherit from *D* by comparing two prime numbers. See our matching algorithms in [14] for more details.

4. Publish/subscribe for low-power μC systems

As computation consumes less energy than wireless communication and to reduce the impact of retransmissions on the energy consumption, it is clear that the most energy savings can be gained by optimizing the wireless communication. Therefore, our μC-SemPS pub/sub subsystem for microcontrollers will favor spending more effort in the semantic event matching if it can further reduce the energy impact of semantic event routing.

Algorithm 1 BuildSubscriptionHierarchy(in: fromPeer, subscription)

```
(subscriptionRelevant, subscriptionForwarded) = (false, false)
if (AlreadyReceived(subscription)) then
   FeedbackSubscription(fromPeer, DUPLICATE, subscription)
else
   MarkReceived(fromPeer, subscription)
if (IsMatchingAncestor(subscription)) then
   subscriptionRelevant = true
else
   LabelSubscription(subscription, OUT_OF_INTEREST)
if (subscription.hopsLeft > 0) then
   subscription.hopsLeft = subscription.hopsLeft - 1
   for each Peer p in ForwardFilter(adjacentPeers, subscription) do
      subscriptionForwarded = true
      ForwardSubscriptions(p, MatchingSubscriptions(subscription))
if (not subscriptionForwarded) then
   if (not subscriptionRelevant) then
      FeedbackSubscription(fromPeer, OUT_OF_INTEREST, subscription)
```

4.1. Semantic event subscription polyhierarchy

The event notification routing protocol aims to reduce the delivery of events to nodes without a matching subscription, as well as the amount of undelivered events for nodes with a matching subscription (i.e. maximize precision and recall). Note that reducing energy consumption does not necessarily mean less messages between peers (as illustrated in Figure 1). The challenge is to organize the peers according to their interests in events (i.e. their subscriptions) to improve the event routing efficiency (both in terms of event delivery accuracy and energy usage for communication).

Our event routing protocol will first establish semantic routing tables by organizing the peers into a subscription polyhierarchy according to the specificity of their subscriptions. A subscription for *PhD Student* events is more specific than a subscription for *Student* events (see Figure 2), which in turn is more specific than a subscription for *Person* events. Note that this example ontology is not that practical for wireless sensor applications, but it is sufficient to the specificity of subscriptions. All the subscriptions of all nodes are classified this way in the subscription polyhierarchy. See Figure 4 for an example of a subscription polyhierarchy of named classes and anonymous classes (intersections of named classes).

Nodes that have a very specialized subscription (e.g. *Foreign male employee*) will be at the bottom of the polyhierarchy, whereas nodes with a very general subscription (e.g. *Female*) are at the top of the polyhierarchy. This means if an event matches one of the nodes in the polyhierarchy, it will also match with a subscription of all the ancestors of the node (due to the specificity relationship). Furthermore, if a node has more than one subscription, it appears multiple times in the polyhierarchy.

A similar classification can be carried out for the publishers by comparing the specificity of the events they produce. If we would include the publishers in the subscription polyhierarchy, then all the descendant subscriptions of a published event would be more specific and hence the produced event would not match. All the ancestor subscriptions of a published event would be more general, and would consume it. However, as the specificity of the produced event is often unknown in advance and typically evolves quicker than the subscriptions, we do not incorporate the publishers in the subscription polyhierarchy.

The algorithm for constructing this polyhierarchy (and semantic routing tables) is partially shown in Algorithm 1. For every subscription a node receives, it basically checks if it already received the subscription via other routes and whether it has a matching local subscription (equivalent or subclass). If so, it forwards its own matching subscriptions to its adjacent peers for which it did not receive a matching out of interest message before. The purpose of the algorithm is to ensure that any event matching the subscription of a node always matches those of its ancestors, and that an out of interest event will not match the subscriptions of the node's descendants.

4.2. Energy based clustering and event routing

In principle, routing events from the root nodes towards the leaf nodes (until they stop matching the subscriptions) is a way to minimize sending events to nodes that do not match their subscriptions. However, there are a few drawbacks with strictly following this routing scheme:

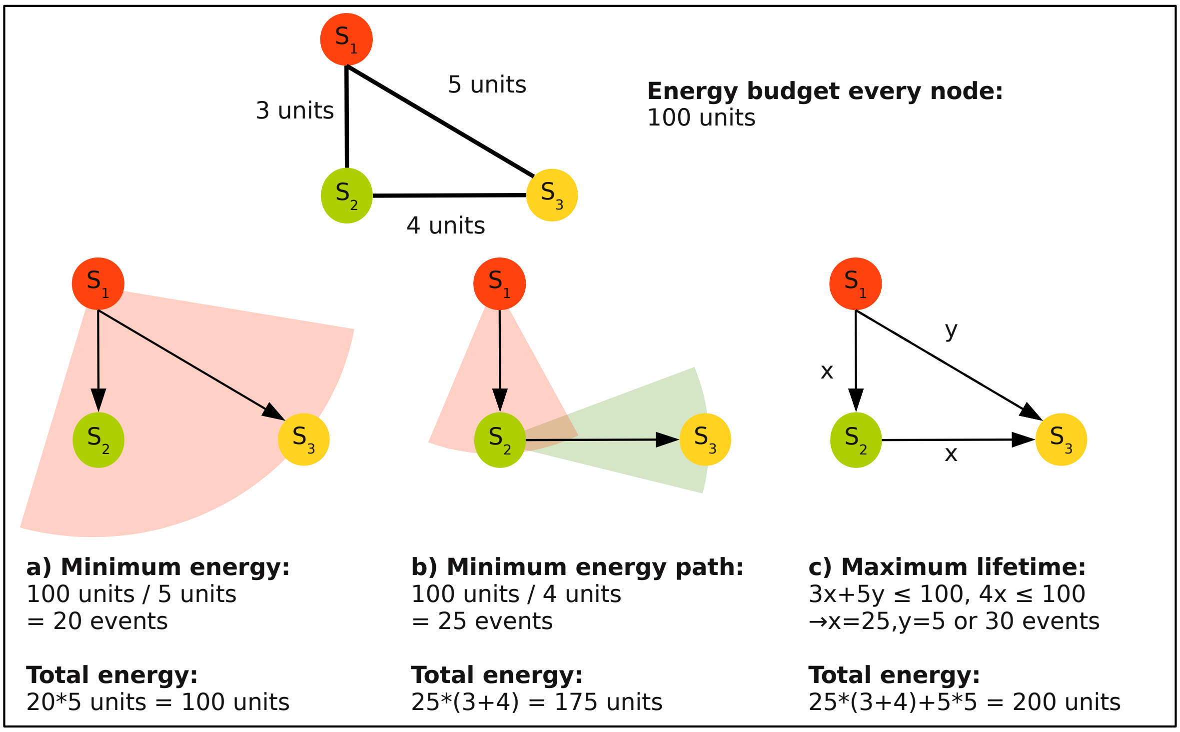

Figure 5. Simplified model for energy usage in different routing scenarios

- The root nodes with broader interests will have a lot more events to forward than the nodes at greater depths in the polyhierarchy.
- Strictly following the routing path may not be efficient from an energy point of view depending on the distance between the nodes involved.

Depending on how we want to optimize the energy consumption, we can define different routing schemes as shown in Figure 5.

This figure shows different scenarios for the case where an event is published by S_1 to which S_2 and S_3 are subscribed:

- (a) This scenario minimizes the total amount of energy by sending the event to all subscribers in one transmission.
- (b) This scenario defines the most energy friendly routing path by minimizing the energy consumption of the transmit amplifier on each node.
- (c) The third scenario attempts to maximize the lifetime of the network by mixing both combinations in order to transmit as many events as possible.

These scenarios are somewhat simplified as they assume communication takes place under ideal operational circumstances. In each of the three scenarios, every node had an interest in the published event. Even in the case where S_2 is not subscribed, these approaches could still be useful to improve the lifetime of the network even if it means a node received and needs to forward an event it is not interested in. However, if none of the subscriptions of the nodes are considered for the routing of the events, we fall back to energy efficient routing protocols for plain communication (LEACH, PEGASIS, TEEN and derivatives [1]). From a pub/sub point of view, such an approach would be wasting energy to routing out-of-interest events, and suffer from a possibly low precision and recall.

Cluster-based data routing [19] is a way to improve energy consumption. To improve traffic confinement we cluster subscribers with similar interests in such a way that a subscriber that receives an event can broadcast it to all the nodes of the cluster it is part of. This guarantees that within a cluster all of the above scenarios can be applied in order to balance the energy consumption over the different nodes in the cluster. It also implies that we need an event delivery protocol to route an event from any publisher to at least one subscriber of every matching cluster.

Our clustering algorithm is based on the same principles of the LEACH protocol in which a percentage P of the nodes (here 5%) become cluster heads:

- Cluster heads are self-selected at the beginning of a cycle. In the r-th cycle, a node that has

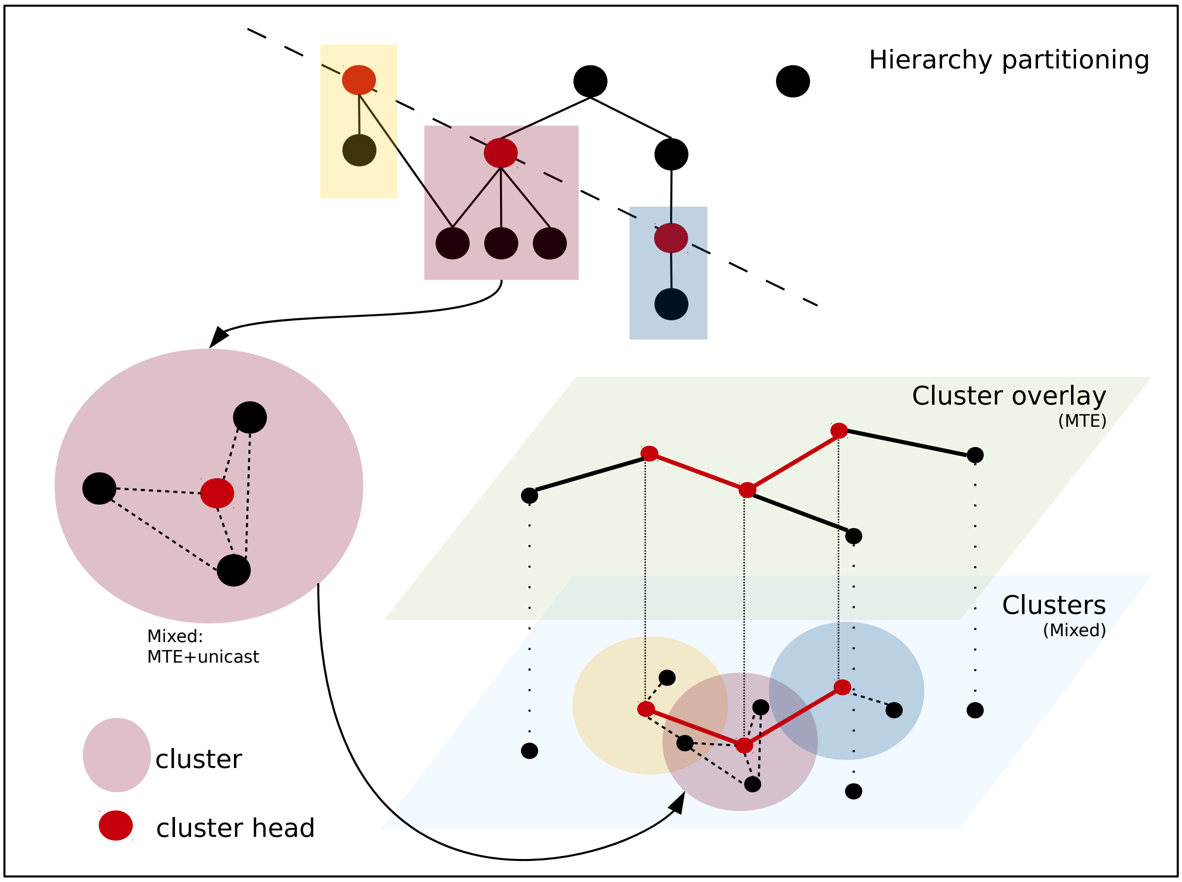

Figure 6. Grouping into clusters of nodes with similar subscriptions

not become a cluster head during the previous 1/P cycles decides to become a cluster head with probability $P/(1-P(r \bmod \lfloor 1/P \rfloor))$

- Other nodes join all clusters for which the cluster head is an ancestor in the subscription polyhierarchy.

The first step rotates cluster heads in such a way that current cluster heads have a lower probability to become cluster heads in the next cycle. The second step groups nodes with similar interests. It is possible that a node does not find a matching cluster, especially if these nodes have subscribed to very general concepts (located near the top in the subsumption polyhierarchy). See Figure 6 for an illustration of this grouping into semantically similar clusters. So rather than having to route events in the network according to subscription polyhierarchy of Figure 4, we can now route on two different levels:

- **Clusters**: within a cluster, we use a mixture of unicasting and minimum transfer energy to maximize the lifetime of the cluster.
- **Cluster overlay**: between cluster heads and nodes without a cluster, we route events with minimum transfer energy (MTE) strategy, because it increases the lifetime better than unicasting and event matches are more likely for nodes with very general subscriptions anyway.

This approach basically breaks down into a partitioning of the subscription hierarchy into two parts: (1) the subhierarchies imposed by the cluster heads and their descendants that will form the clusters, and (2) the hierarchy imposed by the ancestor classes of the cluster heads and the nodes without a cluster. We use two different routing strategies in the two partitions, and by reselecting the cluster heads we balance the energy load amongst all nodes. The efficient encoding and semantic matching makes the overhead of managing the clusters low. This is important for mobile nodes that connect to a variety of nodes and for nodes that frequently change their subscriptions. However, theoretical situations exist where the majority of communication is spent on regrouping messages.

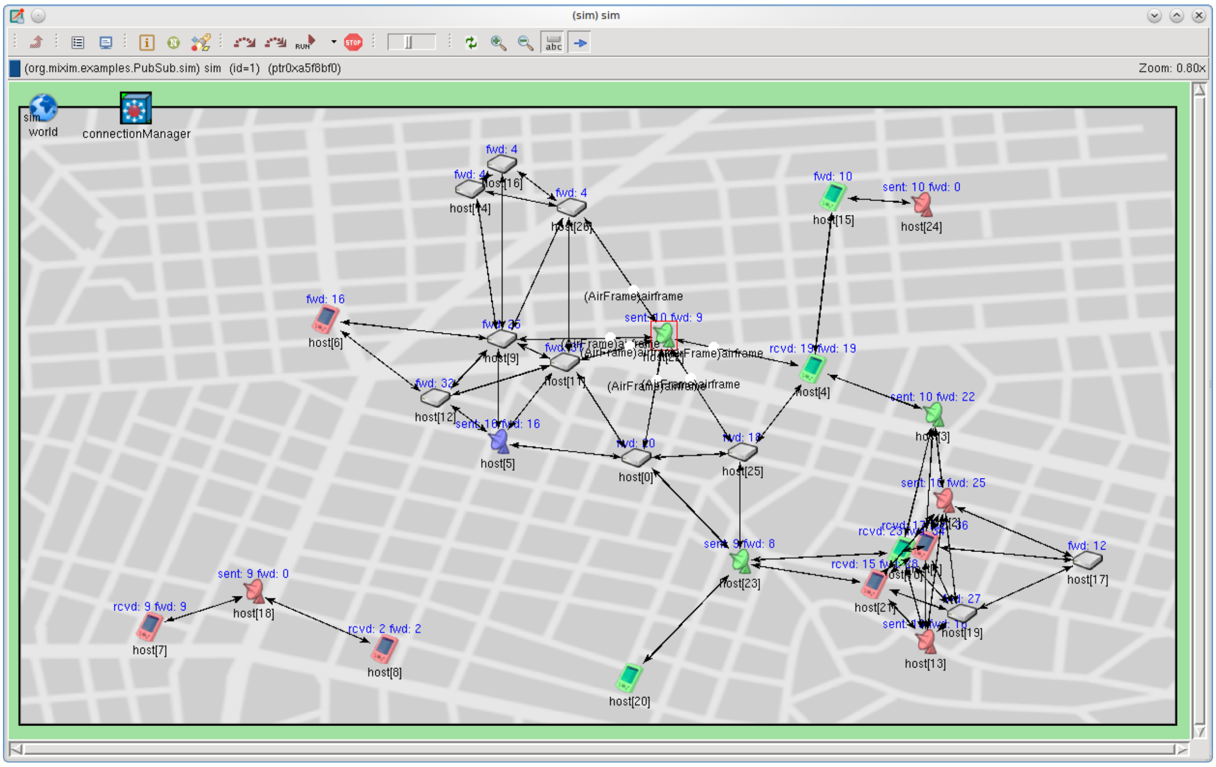

Figure 7. Experimental results with the MiXiM wireless network modeling framework

5. Experimental performance evaluation

In this section, we will analyze both energy usage as well as the publish/subscribe effectiveness of the event delivery (in terms of event routing through non-subscriber nodes). To test the effectiveness of the semantic routing part of our μC-SemPS pub/sub system on larger setups (500 nodes), we conducted simulations with MiXiM[1], an OMNeT++ modeling framework created for mobile wireless and ad-hoc sensor networks with built-in energy framework. We simulated a ZigBee wireless network using energy characteristics of a IEEE 802.15.4 low-power digital radio.

In our 500 node setup, we randomly selected 20 % of the nodes as publishers each producing single events, and another 40 % played the role as subscribers with up to 3 subscriptions. The remaining unselected nodes were merely used for routing purposes. Our implementation supports mobility of nodes and a-periodic event publications, but in order to compare our results with other techniques, we simplified our experimental setup to static ad hoc networks with periodic events. In ongoing work, we are investigating the effects of different topology change rates and packet arrival rates, as well as new decision mechanisms for proactive and reactive routing schemes.

[1] http://mixim.sourceforge.net/

The simulated network depicted in Figure 7 illustrates publishers, represented as antennas, and subscribers represented as personal digital assistants. The color of these symbols represents the type of event that is being published or subscribed to. The network itself is generated by means of a weighted location dependent wiring. For k nodes, the first one n_1 is situated in the center of a unit square. The other nodes n_j, $j = 2...k$, are randomly placed in the unit square and connected to at most m existing nodes n_i that each minimize a different function F_m:

$$F_m(n_i, n_j) = H(n_1, n_j) + w_m.D(n_i, n_j)$$

with H = number of hops to node n_1, D = Euclidean distance, and m different weights w_m = parameter that influences the geographical dependency.

5.1. Energy cost of semantic matching

Regarding the computational complexity, we estimated based on [14] that verifying for a semantic match between two concepts in a very large ontology with more than 25000 concepts requires on average about 50 instructions on an ARMv4 CPU, which at 2 nJ/instruction results in an energy cost of about 100 nJ.

This average was computed by comparing each pair of concepts and verifying whether they semantically match. In case there is no match, we exploit various

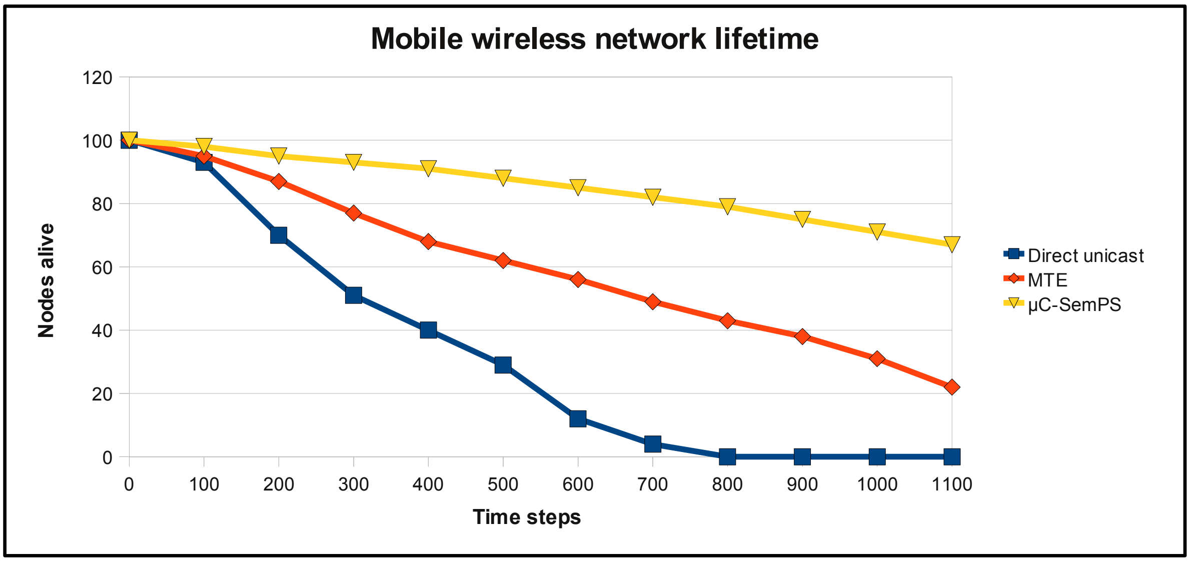

Figure 8. Measuring the network lifetime

efficient heuristics to detect this and only in about 5-10% of the cases we actually need to carry out the more expensive operation to check whether a big integer number can be divided by a prime number. Given that ontologies with more than 1000 unique concepts are rather rare, the cost for semantic matching for a realistic pub/sub application will be lower.

5.2. Energy cost of semantic routing

In Figure 8, you can see the advantage of using our semantic clustering approach with cluster head rotation on the lifetime of the network.

For this particular experiment, the routing protocols from our μC-SemPS semantic publish/subscribe system are more efficient compared to direct unicasting and minimum energy transmission. The energy for computation (the semantic matching) is quite negligible compared to the energy dissipated due to communication (less than 10% for some nodes).

For nodes that were not transmitting themselves, either because they were not a publisher or because they did not forward any event, there was still some communication energy spent for creating and maintaining the clusters, but the significant difference between communication and computation was less outspoken.

Note that these results are very specific for this experimental setup. If the average distance between the nodes grows, then this will have an effect on the total energy dissipation for all protocols.

5.3. Pub/sub efficiency of event routing

We compared in another experiment the efficiency of event routing. In the experiment, we measured how many out-of-interest events were delivered (counting the events received by a node that did not have a matching subscription) and missed events (events that should have been delivered but were not). These metrics boil down to precision and recall from an information retrieval point of view.

$$precision = \frac{\{relevant\ event\} \bigcap \{received\ event\}}{\{received\ event\}} \quad (1)$$

$$recall = \frac{\{relevant\ event\} \bigcap \{received\ event\}}{\{relevant\ event\}} \quad (2)$$

The *precision* metric computes the fraction of the received events that are relevant, i.e. semantically match one of the subscriptions of the node. The *recall* metric computes how many of the relevant events published in the network were successfully received by the node.

A low precision means that a node is spending (and wasting) energy on receiving, processing and forwarding events that do not match any of its subscriptions. A low recall means that a node missed a lot of relevant events. A 100% recall is easy to achieve by making sure every node receives every event (e.g. through broadcasting every received event), but this greatly affects the precision. Obviously, from an energy perspective it would be better if a node did not have to deal with event at all and this would result in

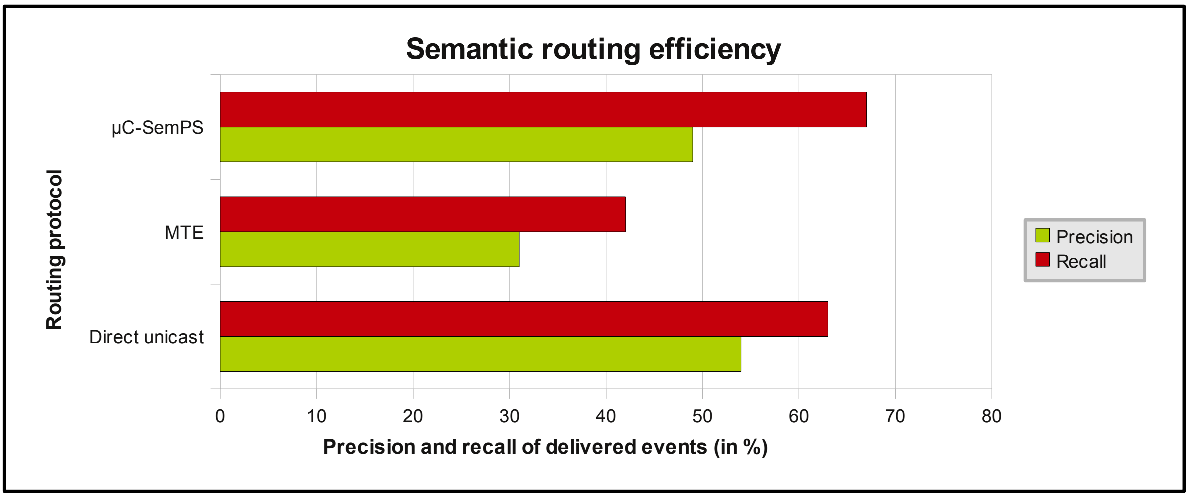

Figure 9. Measuring the delivery of out-of-interest and missed events

an undefined precision and recall, but would also break the purpose of the semantic pub/sub system.

See Figure 9 for the results. Due the fact that MTE uses more intermediate nodes, it is clear that its precision is smaller. The max hop count for delivery also had an effect on its lower recall. The direct unicast performed better, but our μC-SemPS system performed even better due to the semantic clustering approach.

5.4. Bridging the simulation gap

In this section, we briefly discuss the gap that exists when going from simulations to experiments with a prototype implementation on a real test setup. Although we used the MiXiM energy framework of the OMNeT++ discrete event network simulator for wireless and mobile networks to realistically model connections between nodes and interference, there are still some assumptions that we need to revise before we can get an accurate picture of power consumption on a wireless sensor network in the real world.

- Sender and receiver clocks are synchronized with no clock drift
- Nodes can send and receive concurrently, i.e. full-duplex
- Simplified transmission and receiving power mappings for faster simulations
- No interference influences from the environment

Obviously, these assumptions are hard to guarantee in the real world. Time synchronization in wireless sensor networks are a big concern [16] and most of the sensor nodes are half-duplex. To get a very accurate power consumption, we need to consider all the above aspects and have a more realistic power consumption model of the physical layer in the wireless radio [20] than what we have used in our experiments. One possible approach would be to combine the OMNeT++/MiXiM simulations with state-of-the-art wireless sensor emulators so that we have accurate measurements for the overall power consumption – including the energy spent by the CPU – using the native semantic pub/sub code cross-compiled for that particular platform.

However, our goal with μC-SemPS was to have a framework for initial validation of the pub/sub algorithms before trying to implement a prototype for a particular wireless sensor network. Further work on smaller test setups will look into tackling these challenges in order to validate the results of our simulations.

6. Conclusions

As the publish/subscribe community expands towards new computing paradigms like the Internet of Things, it is paramount that semantic awareness and energy consumption is taken into account in the matching and routing of events. While many energy-efficient routing protocols have been proposed for wireless sensor networks, they only focus on optimal delivery paths and do not consider optimizing the routing paths for precision and recall of the pub/sub system.

This article addresses energy-efficient semantic matching and routing for low-power wireless and mobile devices. We presented our μC-SemPS semantic publish/subscribe for battery-powered microcontrollers and sensor nodes. Rather than relying on a centralized broker for ontology-based semantic matching, we use an encoding scheme that represents semantic relationships in an efficient compact representation that helps reduce the packet size of the network messages.

We have demonstrated our clustering technique that takes both semantic similarity as well as energy awareness into account for the efficient delivery of events to subscribers. We have shown that our approach is significantly better in terms of energy consumption and network lifetime while maintaining a high precision and recall of delivered events.

A key direction of our future work, will be to investigate the effects of node mobility and churn to maintain the subscription polyhierarchy, and to create a larger testbed with many more energy-efficient wireless routing protocols. While none of them focus on semantic publish/subscribe systems, it will be interesting to see how well they fare in terms of precision and recall. This could provide interesting insights into how energy efficiency can be traded for more accurate routing to increase precision and recall for the semantic events.

Acknowledgments

This research is partially funded by the Interuniversity Attraction Poles Programme Belgian State, Belgian Science Policy, and by the Research Fund KU Leuven.

References

[1] Akkaya, K. and Younis, M.F. (2005) A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks* **3**(3): 325–349.

[2] Chand, R. and Felber, P.A. (2005) Semantic peer-to-peer overlays for publish/subscribe networks. In *Euro-Par 2005 parallel processing, 30 August - 2 September 2005, Lisbon, Portugal - Also published in LNCS Volume 3648*. doi:10.1007/11549468_130.

[3] Cilia, M., Bornhövd, C. and Buchmann, A.P. (2003) Cream: An infrastructure for distributed, heterogeneous event-based applications. In Meersman, R., Tari, Z. and Schmidt, D.C. [eds.] *CoopIS/DOA/ODBASE* (Springer), *Lecture Notes in Computer Science* **2888**: 482–502.

[4] de Meulenaer, G., Gosset, F., Standaert, F.X. and Pereira, O. (2008) On the energy cost of communication and cryptography in wireless sensor networks. In *WIMOB '08: Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication* (Washington, DC, USA: IEEE Computer Society): 580–585. doi:10.1109/WiMob.2008.16.

[5] Eugster, Felber, Guerraoui and Kermarrec (2003) The many faces of publish/subscribe. *ACM Comput. Surv.* **35**(2): 114–131. doi:10.1145/857076.857078.

[6] Heinzelman, W.R., Chandrakasan, A. and Balakrishnan, H. (2000) Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8* (Washington, DC, USA: IEEE Computer Society): 8020.

[7] Horridge, M. and Patel-Schneider, P. (2008) Manchester syntax for owl 1.1. In *International Workshop OWL: Experiences and Directions (OWLED 2008)*.

[8] I. Demeure et al. (2008) An energy-aware middleware for collaboration on small scale MANets. In *Proceedings of the Autonomous and Spontaneous Networks Symposium (ASNS'08)* (Paris, France).

[9] Jafarpour, Hore, Mehrotra and Venkatasubramanian (2009) Ccd: efficient customized content dissemination in distributed publish/subscribe. In *Middleware '09: Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware* (New York, NY, USA: Springer-Verlag New York, Inc.): 1–20.

[10] Li, S., Lin, Y., Son, S., Stankovic, J. and Wei, Y. (2003) Event detection services using data service middleware in distributed sensor networks.

[11] Ma, J., Xu, G., Wang, J. and Huang, T. (2006) A semantic publish/subscribe system for selective dissemination of the rss documents. In *GCC '06: Proceedings of the Fifth International Conference on Grid and Cooperative Computing* (Washington, DC, USA: IEEE Computer Society): 432–439. doi:10.1109/GCC.2006.19.

[12] Petrovic, M., Burcea, I. and Jacobsen, H. (2003) S-ToPSS: semantic Toronto publish/subscribe system. In *VLDB '2003: Proceedings of the 29th international conference on Very large data bases* (VLDB Endowment): 1101–1104.

[13] Preuveneers, D. and Berbers, Y. (2006) *Prime Numbers Considered Useful: Ontology Encoding for Efficient Subsumption Testing*. Tech. Rep. CW464, Department of Computer Science, Katholieke Universiteit Leuven, Belgium.

[14] Preuveneers, D. and Berbers, Y. (2008) Encoding semantic awareness in resource-constrained devices. *IEEE Intelligent Systems* **23**(2): 26–33.

[15] Shnayder, V., Hempstead, M., Chen, B.r., Allen, G.W. and Welsh, M. (2004) Simulating the power consumption of large-scale sensor network applications. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems* (New York, NY, USA: ACM): 188–200. doi:10.1145/1031495.1031518.

[16] Sivrikaya, F. and Yener, B. (2004) Time synchronization in sensor networks: a survey. *Network, IEEE* **18**(4): 45–50. doi:10.1109/MNET.2004.1316761.

[17] Taherian, S. and Bacon, J. (2007) A Publish/Subscribe Protocol for Resource-Awareness in Wireless Sensor Networks. In *Proceedings of the International Workshop on Localized Algorithms and Protocols for Wireless Sensor Networks* (Springer), *Lecture Notes in Computer Science (LNCS)* **4549**: 27–38.

[18] Wang, Jin and Li (2004) An ontology-based publish/subscribe system. In *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware* (Springer-Verlag): 232–253.

[19] Wang, H.L. and Chao, Y.Y. (2009) A cluster-based data routing for wireless sensor networks. In *ICA3PP '09: Proceedings of the 9th International Conference on Algorithms and Architectures for Parallel Processing* (Berlin, Heidelberg: Springer-Verlag): 129–136. doi:10.1007/978-3-642-03095-6_13.

[20] Wessel, K., Swigulski, M., Köpke, A. and Willkomm, D. (2009) Mixim: the physical layer an architecture overview. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*,

Simutools '09 (ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)): 78:1–78:8. doi:10.4108/ICST.SIMUTOOLS2009.5555.

[21] Zhang, W., Ma, J. and Ye, D. (2008) Fomatch: A fuzzy ontology-based semantic matching algorithm of publish/subscribe systems. *Computational Intelligence for Modelling, Control and Automation, International Conference on* **0**: 111–117. doi:10.1109/CIMCA.2008.111.