μC-SemPS: Energy-efficient semantic publish/subscribe for battery-powered systems

Davy Preuveneers and Yolande Berbers

Department of Computer Science, K.U.Leuven
Celestijnenlaan 200A, B-3001 Leuven, Belgium,
{davy.preuveneers,yolande.berbers}@cs.kuleuven.be,
http://www.cs.kuleuven.be

Abstract. In this paper, we present our lightweight semantic publish/ subscribe system μC-SemPS that is targeted towards battery-powered micro-controllers and wireless sensor nodes in ubiquitous computing environments. The key challenge that we address is to minimize the overall energy consumption for subscription matching and delivery. Our system relies on an efficient representation of semantic subscriptions and favours 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 for wireless mobile networks, show that our approach for routing semantic events significantly reduces energy consumption.

Keywords: energy-awareness, publish/subscribe, semantic routing

1 Introduction

The publish/subscribe (pub/sub) communication paradigm [4] 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 [7] that match and route events/subscriptions to other brokers and clients.

The major challenge that we address in this paper is the energy-efficiency of semantic pub/sub systems for wireless and battery-powered embedded systems. This class of pub/sub systems [2, 10, 3, 9] is gaining importance given the fact that (1) emerging computing paradigms like the Internet of Things 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

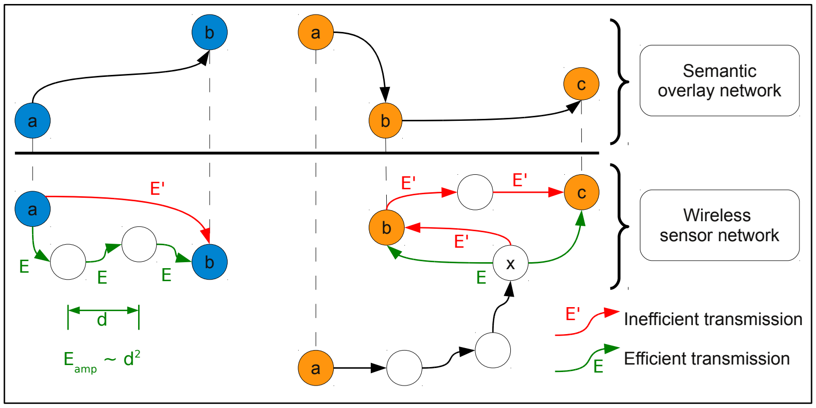

Fig. 1. Mapping of a semantic overlay structure onto a peer-to-peer network

must be unambigously 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 paper, 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. Merely implementing a semantic pub/sub system as a mobile overlay structure on top of a (power efficient) wireless peer-to-peer network can still be inefficient from an energy perspective. A direct unicast between two nodes may be more expensive than sending over intermediate nodes, because the energy consumption of the wireless signal amplifier is quadratically proportional to the distance. This is illustrated in Fig. 1. On the left, 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 decission is not clear cut. If the batteries of the intermediate nodes are almost depleted, a direct unicast can improve the network lifetime. On the right, 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 paper, 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 paper 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

Current solutions attempt to build semantic-awareness with a simple hierarchical topology of concepts (taxonomies of classes and properties). S-ToPSS [10] 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) [14] 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) [9], 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 containment-based and similarity-based proximity metrics [2] for peer-to-peer overlays of pub/sub networks to tackle the lack of expressiveness in many subscription languages.

Many of the above semantic pub/sub systems go beyond the capabilities of embedded devices. The DSWare [8] 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 [13]. 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 [6], 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].

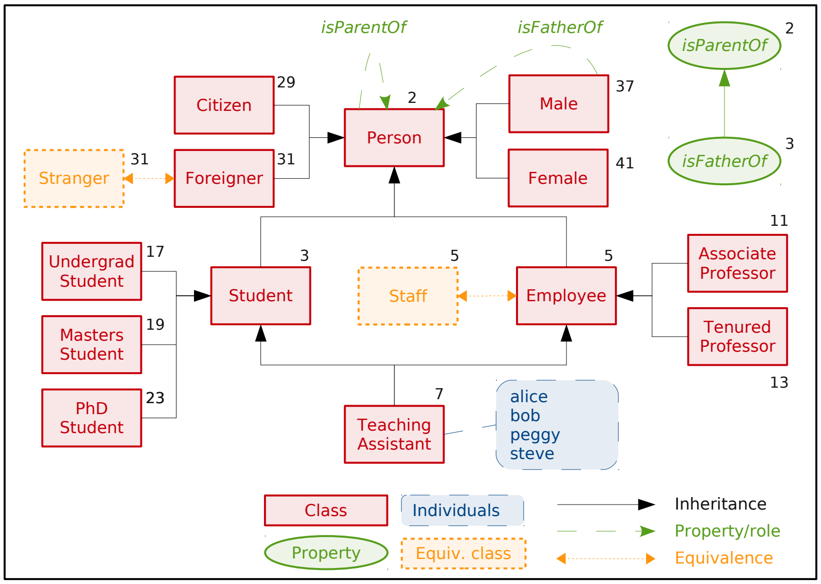

Fig. 2. Inheritance of ontology classes and properties with prime number assignment

3 Semantic event subscription model

In this section, we will briefly discuss the basic concepts of our event and subscription models as used in our μC-SemPS system.

3.1 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 Fig. 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:

1. **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:

$$Student \subseteq Person$$

2. **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):

$$isFatherOf \subseteq isParentOf$$

3. **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*:

$$Car \equiv Automobile$$

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*.

3.2 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 [5] 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.

3.3 Semantic event matching

The encoding that we use in our system counters one of the major disadvantages of semantic matching with ontologies: the reasoning engines supporting the semantic modeling with ontologies are not designed with the limited computational resources of an embedded system in mind. Our prime number-based encoding algorithms were developed with the aim to reduce the memory and processing complexity. See our matching algorithms in [12, 11] for more details.

The encoding of a multiple inheritance hierarchy $\chi = \{Person, Student, Employee, ...\}$ is based on the prime numbers assigned to each class as illustrated in Fig. 2. For example, class *Teaching Assistant* inherits the prime numbers $2, 3$ and 5 from its ancestors and is assigned 7 as its own prime number. The encoding of this class becomes $\gamma(Teaching\ Assistant) = 2 \times 3 \times 5 \times 7 = 210$. As class *Person* has no ancestors, its encoding $\gamma(Person)$ is solely based on its own prime number. Testing for a semantic match can be done with a simple division. For example, *Undergrad Student* is a semantic match for a *Student* because $\gamma(Undergrad\ Student) = 102$ can be divided by the prime number of $Student = 3$. In several cases, the division is not even necessary as many heuristics can be used to rule out subsumption [12, 11]. The degree of compaction solely depends on the prime number assigned to each class. For the very simple hierarchy in Fig. 2, the original OWL ontology file is more than 3000 bytes long,

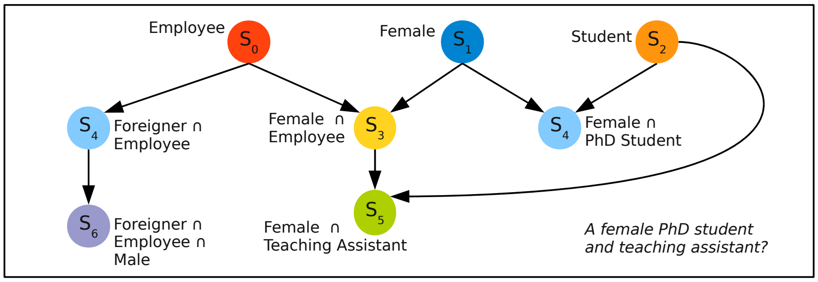

Fig. 3. Organizing nodes according to the specificity of their subscriptions

whereas our prime number encoding generates an equivalent representation of less than 200 bytes. The advantages of our semantic encoding are:

1. A class (or property) can be identified by a simple prime number rather than a possibly long string that represents its URI.
2. All the relevant semantic relationships are embedded in the encoding of each class and property.
3. 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.

4 Publish/subscribe for low-power μC systems

Given that the most energy savings can be gained from optimizing the wireless communication, 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.

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 Fig. 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 polyhierarchy according to the specificity of their

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)
```

subscriptions. Nodes that have an a subscription for a very specialized concept will be at the bottom of the tree, whereas nodes interested in very general concepts are at the top of the tree. If a node has more than one subscription, it appears multiple times in the tree. See Fig. 3 for an example of a subscription polyhierarchy of named classes and anonymous classes (intersections of named classes). 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:

- 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.

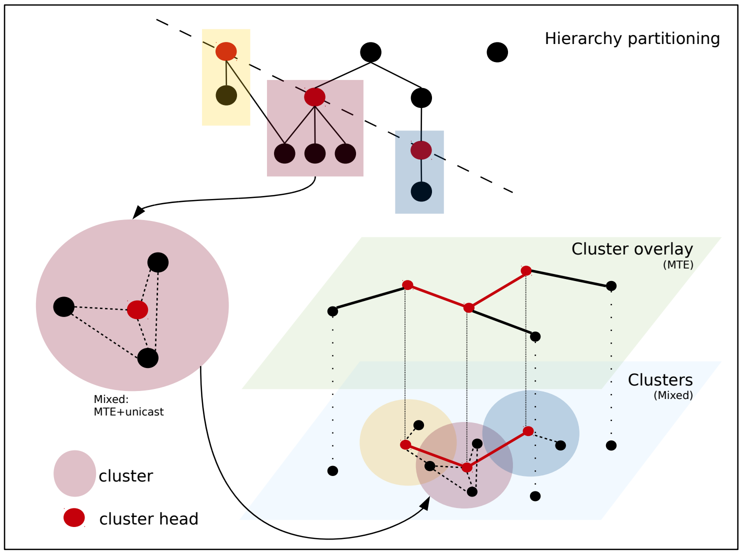

Fig. 4. Grouping into clusters of nodes with similar subscriptions

Cluster-based data routing [15] 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. Our clustering algorithm uses a percentage P of the nodes (here 5%) to become cluster heads:

1. Cluster heads are self-selected at the beginning of a cycle. In the r-th cycle, a node that has not become a cluster head during the previous 1/P cycles decides to become a cluster head with probability $P/(1 - P(r \bmod 1/P))$.
2. 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 Fig. 4 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 Fig. 3, 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.

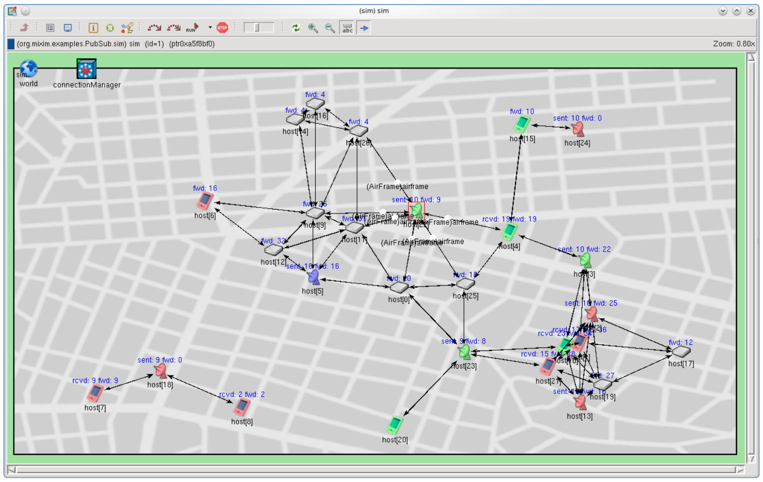

Fig. 5. Experimental results with the MiXiM wireless network modeling framework

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.

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 (see Fig. 5) 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.

5.1 Energy cost of semantic matching

Regarding the computational complexity, we estimated based on [12] that verifying for a semantic match between two concepts in a very large ontology with more

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

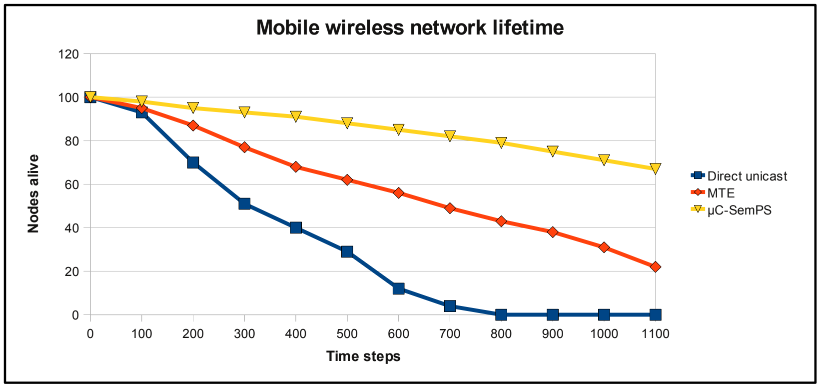

Fig. 6. Measuring the network lifetime

than 25000 concepts requires on average about 50 instructions on an ARMv4 CPU, which at 2 nJ/instruction means something like 100 nJ.

5.2 Energy cost of semantic routing

In Fig. 6, 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 maintaing the clusters, but the significant difference between communication and computation was less outspoken. Note that these results are very specific for the experiment. 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 in which 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. See Fig. 7 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.

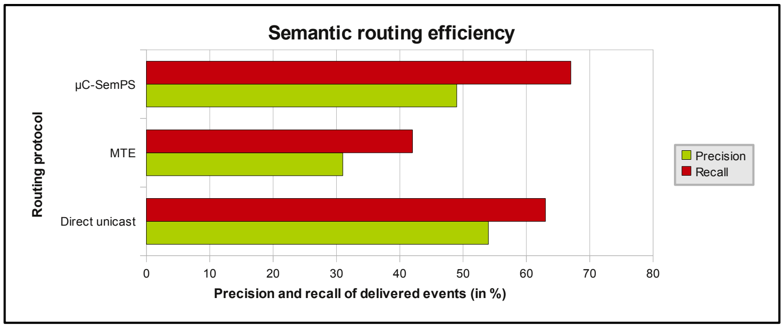

Fig. 7. Measuring the delivery of out-of-interest and missed events

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 paper 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 K.U. Leuven.

References

1. Kemal Akkaya and Mohamed F. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005.
2. Raphael Chand and Pascal A Felber. 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*, 08 2005.
3. Mariano Cilia, Christof Bornhövd, and Alejandro P. Buchmann. Cream: An infrastructure for distributed, heterogeneous event-based applications. In Robert Meersman, Zahir Tari, and Douglas C. Schmidt, editors, *CoopIS/DOA/ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 482–502. Springer, 2003.
4. Eugster, Felber, Guerraoui, and Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.
5. M. Horridge and P.F. Patel-Schneider. Manchester syntax for owl 1.1. In *International Workshop OWL: Experiences and Directions (OWLED 2008)*, 2008.
6. I. Demeure et al. An energy-aware middleware for collaboration on small scale MANets. In *Proceedings of the Autonomous and Spontaneous Networks Symposium (ASNS'08)*, Paris, France, 2008.
7. Jafarpour, Hore, Mehrotra, and Venkatasubramanian. Ccd: efficient customized content dissemination in distributed publish/subscribe. In *Middleware '09: Proceedings of the 10th ACM/IFIP/USENIX International Conference on Middleware*, pages 1–20, New York, NY, USA, 2009. Springer-Verlag New York, Inc.
8. S. Li, Y. Lin, S.H. Son, J.A. Stankovic, and Y. Wei. Event detection services using data service middleware in distributed sensor networks. 2003.
9. JianGang Ma, Gang Xu, JinLing Wang, and Tao Huang. 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*, pages 432–439, Washington, DC, USA, 2006. IEEE Computer Society.
10. M. Petrovic, I. Burcea, and H.A. Jacobsen. S-topss: semantic toronto publish/subscribe system. In *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*, pages 1101–1104. VLDB Endowment, 2003.
11. D. Preuveneers and Y. Berbers. Prime numbers considered useful: Ontology encoding for efficient subsumption testing. Technical Report CW464, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, October 2006.
12. Davy Preuveneers and Yolande Berbers. Encoding semantic awareness in resource-constrained devices. *IEEE Intelligent Systems*, 23(2):26–33, 2008.
13. S. Taherian and J. Bacon. 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*, volume 4549 of *Lecture Notes in Computer Science (LNCS)*, pages 27–38. Springer, 2007.
14. Wang, Jin, and Li. An ontology-based publish/subscribe system. In *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 232–253. Springer-Verlag, 2004.
15. Hao-Li Wang and Yu-Yang Chao. 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*, pages 129–136, Berlin, Heidelberg, 2009. Springer-Verlag.