Message Models and Aggregation in Knowledge Based Middleware for Rich Sensor Systems

Joseph B. Kopena
Drexel University
Philadelphia, PA, USA
tjkopena@cs.drexel.edu

William C. Regli
Drexel University
Philadelphia, PA, USA
regli@drexel.edu

Boon Thau Loo
University of Pennsylvania
Philadelphia, PA, USA
boonloo@cis.upenn.edu

ABSTRACT

Networked, distributed real world sensing is an increasingly prominent topic in computing and has quickly expanded from resource constrained "sensor networks" measuring simple values to "sensor webs" of heterogenous networks encompassing many types of services and hosts, processing a wide variety of data and media. This paper presents ongoing work on OntoNet, which aims to provide messaging middleware in support of such rich sensor systems. In particular, this paper discusses the underlying message delivery model assumptions required in effectively supporting these settings. Those assumptions in turn present large implications for the mechanisms used to describe and match messages and destinations, as well as how to effectively do so in a scalable but correct manner. Initial concepts are also presented for two approaches to aggregating metadata and reducing network and memory consumption in OntoNet. One is a new application of least common subsumer induction, a known but infrequently used description logic inference. The other is a novel application of Bloom filters to representing and querying ontology driven data.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications; I.2.4 [**Computing Methodologies**]: Artificial Intelligence—*Knowledge Representation Formalisms and Methods*

1. INTRODUCTION

Networked, distributed real world sensing is an increasingly prominent topic in computing. The field has quickly expanded from "sensor networks" of low power, wireless, ad hoc networks of resource constrained mote processors measuring simple values [14], to "sensor webs" of heterogenous networks encompassing many types of services and hosts, processing a wide variety of data and media, often over extensive geographic areas [11].

This work is primarily concerned with settings closer to the latter, in which motes and embedded sensors, handhelds, laptops, and desktops or servers, connected via a variety of network mediums, are employed to produce and process a large mix of data, from simple readings and notifications to large data sets, reports, and media. As computing platforms and sensors become more capable, such systems might range from relatively small networks reminiscent of traditional sensor networks, to vast, Internet-enabled, multi-organization sensor webs. There will always be a role for extremely resource constrained, highly focused sensor networks generating and managing continual flows of relatively homogenous data. However, as computing platforms of all sizes become increasingly powerful, batteries longer lived, and available sensors more diverse, there will be increasing focus on complex systems processing an assortment of rich data and media.

Traditional sensor networks are primarily challenged by efficiently collating data at gateways or storing for later collection, all under considerable resource constraints. Whether on sensor webs or smaller networks, rich sensor systems encompassing many types of data producers and consumers don't necessarily suffer the same constraints, but still face a variety of challenges. Critical among these is getting the right data to the right consumers. Given the diversity of sensors and software present and more capable underlying platforms, this problem area is more akin to those of service oriented computing, content based addressing, and peer to peer computing than traditional sensor networking, and is the focus of this work. As opposed to continuous queries over streams of simple measures, the core paradigm is message delivery of larger data and work products such as collections of readings, analyses, and media, as well as requests and notifications.

Presented here is ongoing work on *OntoNet* [12], which aims to provide that service via content delivery and service discovery middleware connecting sensors, services, and agents on the network. OntoNet draws heavily from Semantic Web [5] technology as well as research from sensor, mobile ad hoc, and overlay networking. In particular, this paper discusses the underlying message delivery model assumptions required in effectively supporting these settings. Those assumptions in turn present large implications for the mechanisms used to describe and match messages and destinations, as well as how to effectively do so in a scalable but correct manner.

2. EMERGENCY RESPONSE SCENARIO

Many organizations maintain response teams for chemical, biological, or radiological (CBR) incidents, such as industrial or transport accidents. Unfortunately, current best practices largely use non-integrated and redundant tools applied via manual processes. Even among well provisioned teams, critical data collection is frequently performed via clipboard and pen—a difficult task in full protective suits. Automating these processes stands to significantly improve team performance and efficiency but faces several challenges.

DMSN'09, August 24, 2009, Lyon, France.

Among these are basic network connectivity and management issues. A mix of mobile and fixed nodes may be in use, fielded by a number of different organizations and potentially using low power, low cost radios. Importantly, team members must be able to operate and work together before infrastructure may be deployed, as well as in environments where disconnections are frequent and they must operate independently. These constraints may imply complex, heterogenous network topologies spanning organizations and employing mobile ad hoc and mesh networking techniques as necessary while also utilizing infrastructure such as base stations as available to gain higher performance and wider resource accessibility.

Many information management challenges are also presented by this setting, mostly centered around basic interoperability problems. A wide variety of information may be generated and used by different elements of the system, including command, sensor, and background data, with many different producers and consumers. Several generations of equipment may also be present, from sensors placed years previously to a mix of modern and dated equipment fielded by response teams of varying funding levels. A variety of organizations and manufacturers will likely also have to cooperate, with corporate, local, state, and federal responders all deploying their own equipment, procedures, and policies.

A key aspect of OntoNet's approach is that sensors or software generate messages with associated metadata, with is used to deliver those messages to the appropriate destination(s). This basic functionality can be used to support many tasks such as data and alert dissemination, service requests, and content retrieval. By addressing messages and destinations via declarative, ontology driven languages, the above interoperability challenges may be met with flexible, extensible, formal reasoning mechanisms. Applying that query based messaging paradigm in distributed, robust middleware also decouples message senders and receivers to a great degree, promoting rapid system integration and improved manageability.

3. RELATED WORK

Previous work on *OntoNet* [12] focused on the network protocols enabling metadata dissemination and message delivery. This paper follows on that work from two directions. First is that the assumed messaging paradigm—destinations publishing queries which are matched against message metadata—could not support requests, only announcements. The second came from further study of the proposed hybrid mesh/trees network structure for maintaining destination state and routing messages. Although effective in static settings, the structures performed poorly on mobile networks due to disruption. These factors lead to interest in effective aggregation in support of both requests and announcements, enabling destination state to be spread more widely throughout the network, reducing the network structures required to tightly control its propagation.

Similarly, nearly all existing related messaging middleware, such as publish/subscribe [10, 9] or intentional naming [1, 4], do not recognize the distinction between request and announcement paradigms, and support only one or the other. Other ontology based messaging middleware [16] has also not closely examined the messaging assumptions in use, and frequently applies ontological supports to provide hierarchical topic labels rather than query support.

Several projects [15, 11] exist developing rich sensor systems similar to that envisioned in OntoNet, some also utilizing Semantic Web technology. However, most are based around collecting and fusing data from fixed sensor network installations over the Internet with at least partially centralized repositories and portals. OntoNet instead focuses on smaller, distributed, mobile sensors and users.

```
<cbr2:N4242Spectometry rdf:about="#MSG1134">
  <msg:source>
    <cbr2:Fixed-FT-IR rdf:about="#Sensor03NG">
      <msg:org rdf:resource="&orgs;#NEAir-Haz" />
    </cbr2:Fixed-FT-IR>
  </msg:source>

  <cbr:nuclideDetected>
    <cbr2:Am-241>
      <cbr:confidence>93.0</cbr:confidence>
      <cbr2:concentration>0.002</cbr2:concentration>
    </cbr2:Am-241>
  </cbr:nuclideDetected>
</cbr2:Spectometry>
```

Figure 1: OWL/RDF description of a message containing spectometry data, noting a particular finding.

4. MESSAGE METADATA AND QUERIES

Messages and destinations are described in OntoNet using the Semantic Web's Resource Description Framework (RDF) [13] and Web Ontology Language (OWL) [8]. RDF is an XML syntax for first order models restricted to binary relations—labeled graphs. OWL is an RDF language for encoding ontologies, taken here as formal, machine interpretable, logical specifications of domain structure. OWL is defined by description logic [2], an object oriented subset of first order logic. Both are well documented, open standards endorsed by the World Wide Web Consortium (W3C).

Figure 1 depicts the OWL/RDF description[1] for a typical message in the scenario of Section 2. It declares that the message contains sensor data, the source is a spectrometer deployed by the NEAir-GM organization, contents are in the IEEE/ANSI N42.42 radiation data format[2], and the sensor has detected a particular nuclide[3]. Notably, this description does not replace message contents or formats, such as SensorML, N42.42, CAP, or DoD CBRN in this scenario. Instead it captures metadata and key content, rendering datasets, images, and other opaque messages accessible to general inference.

To receive this message, a process would register an appropriate query defined using OWL class expressions. Figure 2 gives an example that might be registered by an analysis package or handheld display to receive such a message. The class consists of all objects with a source that belongs to the NEAir organization, use the ANSI N42.42 format, and are reporting a suspicious nuclide.

Matching the description against the class requires using OWL semantics to apply background ontologies and derive inferences. Core OntoNet ontologies define generic description elements such as `msg:format` while application specific ontologies encode domain elements and knowledge such as the taxonomy of sensors.

For example, matching Figure 1 and Figure 2 requires knowing and utilizing the background knowledge in Figure 3. The middleware must reason about subclasses, such as `cbr2:Am-241` implies `cbr:SuspiciousNuclide`, apply the `orgs:NEAir-Haz` object's

[1]Namespace declarations, ontology imports, and `rdf:RDF` wrappers have been removed from these examples for clarity.

[2]`http://standards.ieee.org/getN42/`

[3]This example is based on `http://units.nist.gov/Divisions/Div846/Gp4/ANSIN4242/2005/annexC.n42`

```
<owl:Class rdf:about="#Query">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Restriction>
      <owl:onProperty rdf:resource="&msg;#source" />
      <owl:someValuesFrom>
        <owl:Restriction>
          <owl:onProperty rdf:resource="&msg;#org"/>
          <owl:someValuesFrom
            rdf:resource="&orgs;#NEAir" />
        </owl:Restriction>
      <owl:someValuesFrom>
    </owl:Restriction>

    <owl:Restriction>
      <owl:onProperty
          rdf:resource="&cbr;#nuclideDetected" />
      <owl:someValuesFrom
          rdf:resource="&cbr;#SuspiciousNuclide" />
    </owl:Restriction>

    <owl:Restriction>
      <owl:onProperty
          rdf:resource="&msg;#format" />
      <owl:hasValue
          rdf:resource="&nist-ansi-n4242xsd;#" />
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Figure 2: OWL query for reports of suspicious nuclides in N42.42 format from a particular organization.

membership in `orgs:NEAir`, and infer a value for the `msg:format` property via explicit membership in `cbr2:N4242Spectrometry`. Such inference and deduction of implicit information promotes interoperability, system evolution, and byte savings. It also differentiates the ontological, knowledge based approach from relational database, XML, and other data oriented techniques.

5. MESSAGE MODELS

Taken as given in Section 4 is a core assumption of message metadata being a description—instance data about an object—which is matched against queries specified by destinations. That such an assumption is being made is often overlooked in such middleware and is frequently implicit in the matching approach employed. However, the chosen directionality of the matching has large implications for the extensibility and interoperability of applications constructed on the middleware. This is particularly important for sensor systems such as in Section 2, where many heterogenous components from a variety of vendors are fielded by multiple organizations, potentially alongside legacy components.

OntoNet's matching model as presented in Section 4 has three components: Messages, destinations, and ontologies. Each message m is associated with a message object m' and description d in relation M. Each destination process p is associated with at least one query q in the relation D. There is also a set of known or retrievable background ontologies B. For a multicast message to be delivered to all matching processes, denoted by *dest*, the matching model is then:

$$\forall\,(m,m',d)\in M,\ (p,q)\in D$$
$$\left[d \bigwedge_{b\in B} b \models q(m')\right] \Rightarrow (m,p)\in dest$$

```
<owl:Class rdf:about="&cbr2;#N4242Spectrometry">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="&msg;#Message" />

    <owl:Restriction>
      <owl:onProperty rdf:resource="&msg;#format" />
      <owl:hasValue
        rdf:resource="&nist-ansi-n4242xsd;#" />
    </owl:Restriction>
  </owl:intersectionOf>
  <rdfs:subclassOf rdf:resource="">
</owl:Class>

<orgs:NEAir rdf:about="&orgs;#NEAir-Haz" />

<owl:Class rdf:about="&cbr2;#Am-241">
  <rdfs:subclassOf
    rdf:resource="&cbr;#SuspiciousNuclide" />
</owl:Class>
```

Figure 3: Excerpts of ontologies required to match the description in Figure 1 against the query in Figure 2.

Entailment in that model is defined by the semantics of the logic used, in this case description logic.

Alternatively, each message could be associated with a query, and each destination with a description:

$$\forall\,(m,q)\in M,\ (p,p',d)\in D$$
$$\left[d \bigwedge_{b\in B} b \models q(p')\right] \Rightarrow (m,p)\in dest$$

The former specifies *receiver querying*, in which destinations publish queries that are matched against message descriptions. Its converse is *message querying*, in which message metadata is a query matched against destination descriptions. Complicating middleware for these rich sensor systems, there are clear roles for both approaches. Distinguishing uses for each are largely centered around publishing notifications versus issuing requests.

Among the background statements in Figure 3 is a very small part of a taxonomy of nuclides used in the example scenario. It asserts that a particular Americium nuclide, `cbr2:Am-241`, is one of a class of nuclides of particular interest, `cbr:SuspiciousNuclide`. Using receiver querying, this knowledge can be used to connect, for example, an alert system watching for messages reporting suspicious nuclides, and a sensor generating reports about Americium detections. This is exactly the example of Figures 1 and 2: The receiving alert system queries for `cbr:SuspiciousNuclide` while the sensor generates messages annotated to report `cbr2:Am-241`, which are correctly inferred to match the query.

Under a similar message querying arrangement, the correct inference is not possible. A description for the alert system listing it as interested in `cbr:SuspiciousNuclide` will not match a more specific query attached to a message for `cbr2:Am-241`. This could be addressed by the alert system advertising every nuclide it understands or the message being annotated with less specific metadata, but those present clear negative ramifications for future extensibility and interoperability of the system. The primary attraction of the ontology driven, knowledge based approach taken in OntoNet

is precisely not requiring such enumeration or loss of specificity.

Unfortunately, there are similar cases with the opposite behavior. For example, a user or automated system may issue a request for a mobile sensor (human carried or unmanned) that can specifically detect Americium to sample an area and confirm a reading. On the same network, a sensor may post that it can detect suspicious nuclides, but not necessarily Americium. Under a message querying approach, the sensor would be correctly skipped as it does not necessarily provide the required more specific functionality. Conversely, under receiver querying, the request would be incorrectly delivered to the sensor as `cbr2:Am-241` implies `cbr:SuspiciousNuclide`, for which the sensor has a general capability but may not necessarily be able to support that particular request. A human user may be able to sort out the mismatch and not select that sensor service, but it impairs efficiency and would be challenging for automated systems to resolve.

In this case it may be reasonable to resolve the issue by requiring the sensor to post a more specific query, but that is not a general solution. On one hand, that requires services to be accurately modeled by their associated metadata or incur problems and inefficiences, while under the query modeling approach the sensor would simply not match. On the other hand, it is simple to shift the example and reintroduce the problem, e.g., by the sensor advertising capability to detect Americium but the user requesting a specific Americium nuclide (e.g., ^{241}Am versus ^{242}Am).

Further, there are many easily conceived scenarios in which rich sensor systems may require middleware support for both messaging models. Another example is publishing infrared imagery—which can likely be utilized by a number of generic imagery consumers—versus requesting infrared imagery—which cannot necessarily be provided by just any imagery provider such as a standard camera.

As demonstrated by these examples, the distinction between the two paradigms is largely that between publishing and requesting:

- Receiver querying is not suitable when more specific destinations can and should process the message, e.g., an infrared camera answering a general request for imagery. It is most suitable if the receiver cares about precisely what class of messages it receives, e.g., requiring a specific format or message type, but the sender does not. This aligns with publishing information to be consumed by any interested entity.
- Message querying is not suitable when more general destinations can and should process the message. It is most suitable if the sender cares about precisely what class of destination receives the message. This aligns with issuing a request that may only be filled via services with certain capabilities.

These notions do not apply where message and destination metadata must match exactly, as in topic based publish/subscribe, but hold up when applied to many existing middleware systems. For example, intentional naming [1, 4] applies message querying to enable dispatching requests to services capable of fulfilling specific criteria. Conversely, publish/subscribe systems based on hierarchical topics or XML and other content filters [10, 9] provide receiver querying to deliver published items to any relevant subscriber.

6. DISSEMINATION AND AGGREGATION

Messaging middleware comprises three closely entwined elements:

- Reasoning and matchmaking to determine proper destinations for generated messages, as discussed in Sections 4 and 5.
- Transportation of messages about the network and delivery to end destinations, providing reliability, QoS, or other traits.
- Mechanisms for disseminating and maintaining destination advertisements and/or queries throughout the network to match against message metadata and guide transport.

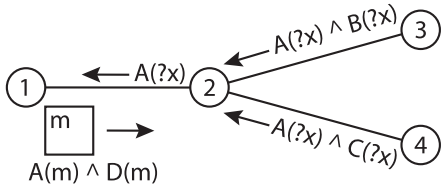
Trivial approaches to disseminating destination metadata are relying on centralized registrars or broadcasting to all nodes. Centralized registrars are problematic for many reasons, including performance bottlenecks and critical failure points. Distributed approaches are clearly warranted for wireless, mobile, ad hoc systems such as that in Section 2 in order to manage in such dynamic networking environments prone to faults and partitions. But even infrastructure and Internet based sensor webs may require distributed approaches to meet performance and robustness requirements as well as to operate across organizational and physical boundaries. Although perhaps not to every node, destination metadata must be disseminated throughout the network in some fashion in order to properly guide messages to their correct destinations.

Given their close logical relationship, ideally both receiver and message querying may be supported by the same mechanisms for propagating metadata and matching messages. This is not feasible for some middleware. There is no easy way to convert most XML publish/subscribe systems [9] to a message querying paradigm because the destinations subscribe via queries directly over published data, messages are not associated with metadata. In other cases this shift is possible. Intentional naming in the INS [1] could likely be readily extended to also support receiver querying as the query and description languages are (nearly) equivalent, and all destination metadata is disseminated to and stored at all nodes. Which matching paradigm to use would be a relatively simple matter of extending the API, effectively marking destinations as either subscribers or services and messages as either publications or requests.

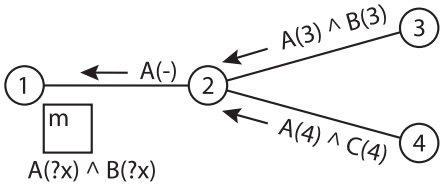
Similarly, OntoNet's RDF/OWL based metadata language and matching logic is easily applicable to either message or receiver querying. This is complicated, however, in trying to reduce the amount of state propagated through the network and stored at each node, an important aspect of improving system scalability. Aggregating or otherwise condensing destination metadata typically entails a reduction in specificity in the form of more permissive queries or lost details, resulting in either false positives or false negatives.

Figure 4 depicts this problem for the two message models. Two nodes are publishing available services while a third is generating messages. An intermediary node on the network aggregates the destinations' metadata to conserve network or memory resources before forwarding on to the sending node. For receiver querying, as in Figure 4a, this potentially produces false positives. Unwanted messages are incorrectly forwarded through the network because the combined query is overly permissive. In this case the message may be dropped at the intermediary node provided it has kept full queries, but unnecessary traffic will still have been generated along the link, path, or region between it and the originator.

Of more concern, applying similar aggregation in a message querying approach may produce false negatives, incorrectly dropping messages as in Figure 4b. In the rich sensor systems of interest here, the possibility of erroneously halting potentially critical re-



(a) Receiver querying: Message m is incorrectly forwarded.



(b) Message querying: Message m is incorrectly dropped.

Figure 4: Demonstration of false positives and negatives produced by aggregating destination metadata under the two message models. In both cases nodes 3 and 4 are potential destinations whose metadata is aggregated at node 2 and advertised to the sender at node 1.

quests or updates is a clear problem. This relationship between information loss and opposing behavior of the two message models—false positives versus false negatives—also implies that it may be difficult or even impossible to apply the same machinery for practical, scalable matching and forwarding to support both paradigms.

Toward addressing these issues and scalably supporting both receiver and message querying in rich sensor systems, two separate aggregation approaches are being investigated for OntoNet. One, presented in the following subsection, applies an existing non-standard description logic inference to optimize receiver queries. The other, outlined in Section 6.2, introduces a new mechanism for aggregating ontology based data in support of message querying.

6.1 Least Common Subsumer

One approach being investigated in OntoNet for receiver querying aggregates destination metadata via inferring the least common subsumer (LCS) [3] of their advertised queries. LCS inference applies description logic semantics to rewrite a set of class definitions into a single class definition of which each input is a subclass, and for which no other such class up to equivalence may exist in the given logic. It is closely related to relational query optimization.

For example, the destination query in Figure 2 may be written in description logic notation as:

$$\begin{aligned}\text{Query1} \equiv\ & \exists \text{msg:source} \,.\, [\exists \text{msg:org} \,.\, \text{orgs:NEAir}] \sqcap \\ & \exists \text{cbr:nuclideDetected} \,.\, \text{cbr:SuspiciousNuclide} \sqcap \\ & \exists \text{msg:format} \,.\, \{\text{nist-ansi-n4242xsd}\}\end{aligned}$$

A similar but different query used on another nearby host may be:

$$\begin{aligned}\text{Query2} \equiv\ & \exists \text{msg:source} \,.\, [\exists \text{msg:org} \,.\, \{\text{orgs:NEAir-Haz}\}] \sqcap \\ & \exists \text{cbr:nuclideDetected} \,.\, \text{cbr:IndustrialNuclide} \sqcap \\ & \exists \text{msg:authentication} \,.\, \text{msg:X509Certificate}\end{aligned}$$

The LCS of those two queries would be:

$$\begin{aligned}\text{Query3} \equiv\ & \exists \text{msg:source} \,.\, [\exists \text{msg:org} \,.\, \text{orgs:NEAir}] \sqcap \\ & \exists \text{cbr:nuclideDetected} \,.\, \text{cbr:Nuclide}\end{aligned}$$

LCS inference may be used to summarize multiple destination queries into a single less discriminatory but smaller query disseminated to remote parts of the network, reducing propagation and state costs. Actual LCS reasoning itself is well understood for several description logics of limited expressiveness, such as $\mathcal{EL}$, which are used in OntoNet due to theoretical and practical complexity concerns.

Challenges. The challenge in this aggregation approach is balancing false positives and consequent unnecessary traffic against bandwidth and memory consumption. Effectively applying this technique requires both efficient protocols for propagating performance data such as false positive rate upstream, as well as algorithms to use that feedback in controlling the aggregation. Such feedback protocols may create large amounts of network traffic overhead. Evaluating the tradeoff between the resources consumed by the metadata propagation versus the consequent false positive rate may also be computationally expensive, as it may require generating and storing aggregations of varying combinations of subsets of metadata known to the propagating host.

6.2 Bloom Filter Encoding

The second approach being developed for aggregation in OntoNet supports message querying. Destination descriptions are summarized in Bloom filters [6], constant length bit strings, encoding the set of tuples implied by the description logic semantics. This is a novel application of Bloom filters to ontologies, though it is similar to uses in relational databases as well as P2P applications [7].

Table 1 outlines the algorithm for encoding a description. Once encoded, filters may then be propagated through the network to provide a constant length description of available services. Querying to match messages with services proceeds similarly to encoding, but checking bits instead of setting them. Notably, the encoding makes aggregations of multiple services, either on a host or across a network region, straightforward to compute and of finite length as they are simply the logical disjunction of the bit strings.

Challenges. Of course, this approach discards much information and suffers several significant limitations. Among these are:

- There is no way to both aggregate the filters and apply them as queries, so only message querying is supported.
- Only homed, connected objects [16] may be connected and queried—there must be a given starting object where encoding and matching begins. However, this aligns well with the task at hand of describing and matching against destinations.
- Only relatively inexpressive query logics may be used. For example, it is not possible to apply description logic all-values restrictions. Simpler logics such as $\mathcal{EL}$ that work

CreateFilter(OWL ontologies, OWL description, URI root)
begin
tuples kb ← DLForwardFixpoint(ontologies, description);
filter ← new bit[];
HashObject(kb, filter, root, 0);
return *filter*
end

HashObject(tuples kb, bit[] filter, URI root, integer offset)
begin
foreach $t \in \Pi_{object}(\sigma_{predicate="rdf:type" \wedge subject=root}(kb))$ **do**
hash t and mark appropriate bits in filter;
end
foreach $(p, o) \in \Pi_{predicate, object}(\sigma_{predicate \neq "rdf:type" \wedge subject=root}(kb))$ **do**
offset = hash p;
HashObject(kb, filter, o, offset);
end
end

Table 1: Algorithm sketch for constructing Bloom filter given background ontologies, destination description, and identifier.

within this approach may however be more than sufficient for many applications.

- New inferences cannot be drawn from the received filters. The originating node must have and apply all background ontologies that might be used by receiving nodes querying the filter. This is a notable limitation on system extension and interoperability, but may be acceptable in many settings if global ontology sharing is feasible.

Despite these limitations, this approach shows potential for a constrained, easily aggregated representation of message query destinations suitable for efficient propagation and storage throughout the network. Importantly, it also eliminates false negatives potentially associated with message querying as discussed above. Instead, as with all practical Bloom filter applications, encoding and aggregation may introduce false positives, a much less critical shortcoming.

Determining the feasibility of this approach is ongoing work in OntoNet, in particular estimating filter sizes required for realistic settings. Protocols and mechanisms for propagating filters and performing forwarding decisions are also in development.

7. CONCLUSION

This paper has described messaging models required to support rich, mobile sensor systems. Although simple, the distinction between announcements and requests and their correlation to message matching models is overlooked in most middleware systems, with typically only one or the other supported. Also presented were approaches to aggregating metadata in support of both message models, including introduction of a novel application of Bloom filters to approximate encoding and querying of ontology based data. Future work in OntoNet is focused on implementing these aggregation procedures and applying them on extremely simple, robust network structures that trade additional state and memory consumption for reduced network traffic and improved disruption tolerance. An important related line of work is development of realistic ontologies and workloads to enable live trial demonstration use as well as evaluation of both aggregation and network procedures.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants 0819845 and 0820208. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley. The design and implementation of an intentional naming system. In *ACM Symposium on Operating Systems Principles*, pages 186–201, 1999.

[2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook*. Cambridge Univ. Press, 2003.

[3] F. Baader, R. Kusters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In *International Joint Conference on Artificial Intelligence*, pages 96–101, 1999.

[4] M. Balazinska, H. Balakrishnan, and D. Karger. INS/Twine: A scalable peer-to-peer architecture for intentional resource discovery. In *International Conference on Pervasive Computing*, pages 195–210. Springer, 2002.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):28–37, 2001.

[6] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[7] A. Broder and M. Mitzenmacher. Network applications of Bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2004.

[8] M. Dean, G. Schreiber, et al. OWL Web Ontology Language reference. World Wide Web Consortium (W3C), February 2004. `http://www.w3.org/TR/2004/REC-owl-ref-20040210/`.

[9] Y. Diao, S. Rizvi, and M. Franklin. Towards an Internet-scale XML dissemination service. *International Conference on Very Large Databases*, pages 612–623, 2004.

[10] P. T. Eugster, P. A. Felber, R. Guerraoui, and A. M. Kermarrec. The many faces of publish/subscribe. *ACM Computing Surveys*, 35:114–131, 2003.

[11] P. B. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan. IrisNet: An architecture for a worldwide sensor web. *IEEE Pervasive Computing*, 2(4), October–December 2003.

[12] J. B. Kopena and B. T. Loo. OntoNet: Scalable knowledge based networking. In *4th International Workshop on Networking Meets Databases*, 2008.

[13] O. Lassila, R. Swick, et al. Resource Description Framework (RDF) Model and Syntax Specification. World Wide Web Consortium (W3C), 1999. `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`.

[14] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. In *5th Symposium on Operating Systems Design and Implementation*, pages 131–146, December 9–11 2002.

[15] A. Sheth, C. Henson, and S. Sahoo. Semantic sensor web. *IEEE Internet Computing*, pages 78–83, 2008.

[16] J. Wang, B. Jun, and J. Li. An ontology-based publish/subscribe system. In *International Middleware Conference*, pages 232–253, Toronto, Canada, 2004.