Mobile Ad Hoc Services: Semantic Service Discovery in Mobile Ad Hoc Networks

Andronikos Nedos, Kulpreet Singh, and Siobhán Clarke

Distributed Systems Group, Trinity College Dublin, Ireland

Abstract. Mobile ad hoc networks (MANETs) are a class of networks where autonomous mobile devices with wireless communication capabilities cooperate to provide spontaneous, multi-hop connectivity. The opportunistic and dynamic characteristics of these networks make discovery of services difficult as they preclude the use of agreed, predefined service interfaces. Using semantic services and permitting their description with multiple domain ontologies is more realistic in this environment because it increases service expressiveness and does not require consensus on a common representation. However, the techniques used in resource-rich, globally connected environments to relate different ontologies and discover semantic services are inappropriate in MANETs. We present here a model for semantic service discovery that facilitates distributed ontology matching and provides scalable discovery of service provider nodes. It uses a gossip protocol to randomly disseminate ontology concepts and a random walk mechanism to identify candidate providers. The model requires no central coordination and the use of randomisation gives it good scalability properties.

1 Introduction

Mobile ad hoc networks are composed of autonomous, wireless nodes which act as mobile routers to provide communication without the need for fixed infrastructure. These networks have an opportunistic aspect as they can form anywhere with little or no coordination. The resulting unpredictability is both appealing and problematic. While these networks require no existing infrastructure, their decentralised topologies make resource identification challenging. In this paper we examine the issue of service discovery in MANETs given the assumption of service role symmetry in nodes, that is, each node has the potential to be both a service provider and a consumer.

Current service description and discovery mechanisms in MANETs [1,2], rely on standardised interfaces to achieve the necessary consensus that makes advertising and discovery possible. While this is usually sufficient in environments that are centrally administered, it poses a serious problem for serendipitous application interoperability in open, distributed systems. For uncoordinated service interaction in MANETs, service based applications would benefit from an expressive service specification language, increased autonomy in the description of individual services and distributed discovery mechanisms.

The use of ontologies in the description of services has already been proposed as a solution for more flexible discovery [3,4]. However, most semantic service description languages and architectures require a globally connected network such as the web. In ad hoc networks, assumptions of global connectivity do not hold because the properties

A. Dan and W. Lamersdorf (Eds.): ICSOC 2006, LNCS 4294, pp. 90–103, 2006.

of the environment are drastically different. One core difference in our model is that nodes do not share a common semantic representation for their services but instead it is assumed that semantic agreement can be derived through node interaction. We believe that this a more appropriate assumption that is in line with the network's uncoordinated and spontaneous formation. Violating this assumption would require either one global ontology to describe all services or maintenance of all potential ontologies in each node. The former is restrictive while the latter is not practical. The idea of semantic decentralisation is also present in decentralised P2P networks [4,5] but has not yet appeared in the domain of mobile ad hoc networks. Below we present a summary of MANET characteristics that form the assumptions of our proposed model:

- Nodes can be both providers and consumers of services. This symmetry precludes the use of centralised brokering architectures. Instead, discovery facilities should be distributed across the participating mobile nodes.
- Nodes are autonomous and consist of small, low-powered devices that can belong in different administrative domains. The implications of this are that applications will be coming from different sources and can be developed independently without a central distribution channel.
- Connectivity is local to the network with no guarantees for Internet availability. This characteristic enforces designs where applications are both self-contained and reactive. Applications will need to function in the absence of any service but also be able to discover and use any desired services as provider nodes connect to the network.
- Connectivity is also intermittent with nodes appearing and disappearing at any moment. Assuming that participating nodes have similar processing capabilities, any load imposed by service discovery has to be shared uniformly. Increasing the processing overhead in select nodes is not desirable and it also leads to extra protocol complexity caused by node election and mandatory fault-tolerant behaviour.

The contribution of this paper is the description of a network model for the discovery of autonomous semantic services in MANET environments. An evaluation based on simulation demonstrates good scalability characteristics as the number of nodes and ontology sizes increase. Although this paper is focused on the network support for the discovery part, the tight dependency to a gossip protocol that is used to facilitate discovery and match heterogeneous ontologies necessitates a brief description presented in section 2.2.

The paper is organised as follows. Section 2 introduces the model and the underlying gossip-based protocol. Section 3 describes how we specify services and discusses the ontology matching algorithm. In Section 4 we present the random walk protocol for semantic service discovery in MANETs. Section 5 presents the evaluation results while we conclude with the state of the art and final remarks in Sections 6 and 7.

2 A Model for Semantic Service Discovery in MANETs

Recent standards for semantic web services, e.g., WSMO, OWL-S, WSDL-S, specify how services are described but are independent of any concrete mechanisms for the discovery of services. To enable the use of such standards in an environment where

provider nodes are autonomous and no fixed infrastructure exists, we have identified two problems that discovery has to address:

Discovery queries must be interpreted by nodes with heterogeneous ontologies – So far a single domain ontology has been assumed for the description of MANET services. This makes possible the direct evaluation of discovery queries between different mobile nodes. We argue that given the opportunistic and unpredictable interaction patterns in MANETs, it is inappropriate to assume that a global ontology is available in every mobile node. Rather, each autonomous node will maintain its own ontology to describe its own services. Since a shared understanding is still required for meaningful semantic interpretation and service interaction, an ontology matching process is needed.

Lack of persistent and centralised service registries – In connected, fixed networks, certain assumptions can be made about longevity of ontology references. Usually, a URI reference is enough for clients to obtain an ontology from a central repository. Resource availability and infrastructure support in these environments also means that ontology matching and semantic service discovery can employ sophisticated techniques that make use of central facilities. In MANETs, transient communication renders brokering architectures harder to support. In addition, the self-contained nature of ad hoc networks means that only ontologies in the participating nodes can be used.

Part of the proposed solution is based on a gossip protocol that exchanges randomised subsets of concepts between nodes. On each node, received concepts are stored in a buffer until certain conditions are met. As a consequence, this buffer holds a constantly evolving and randomised set of concepts. A lightweight ontology matching mechanism in each node matches received concepts with those stored in the node's buffer. We consider this random set of concepts maintained by each node to be a view on all ontologies that has the following properties: It is *partial*, since no view contains the union of all concepts of all ontologies; it is *evolving*, the gossip protocol constantly inserts and removes concepts from this view while the matching algorithm in each node establishes associations between received and stored concepts; it is *randomised* since each view does not contain a set of concepts that concretely describe a knowledge domain, rather it contains randomised concepts that can belong to any of the available ontologies.

2.1 System Model

We consider an ad hoc network $N = \{n_1, n_2, \cdots, n_m\}$ as a set of mobile nodes $n_i \in N$ of size m. Each mobile node is considered to be an active participant that provides services described by an ontology. Each node maintains three different views. The *ontology view*, the *concept view* and the *node view*. We denote each of the three views at node n_i as V_i^O, V_i^C and V_i^N correspondingly.

The ontology view represents an ontology in each node as a fixed set of concepts, $V_i^O = \{c_{i1}, \cdots, c_{ig}\}$. Here, $c_{il} \in V_i^O, 1 \leq l \leq g$ is a concept in the ontology of source node n_i while g represents the maximum number of concepts. We assume that these ontologies are static so this view allows no additions or deletions of concepts for the duration of the protocol execution. This assumption is reasonable as ontologies are structured metadata, meaning they are specified during application design and don't change often. In contrast, services are described as metadata instances and their description can change at any time.

The concept view includes the set of concepts that are received from other nodes and does not allow duplicate concepts or concepts that exist already in the same node's ontology view. We represent this view as $V_i^C = \{c_{kl} \mid c_{kl} \in V_k^O, k \in N - \{n_i\}, 1 \leq l \leq g\}$, where c_{kl} represents any concept from any ontology view other than the one in node n_i. The concept view has variable size, i.e., concepts are added and removed during the execution of the protocol but the gossip protocol guarantees that V^C only contains a subset of the overall concepts.

The node view is composed by a set of node identifiers, $V_i^N = \{n_k | k \in N - \{n_i\}\}$. It maintains a uniform, randomised, partial and fixed-size set of node ids and is populated during a bootstrap phase. Like the concept view it does not allow duplicate node identifiers and does not contain the node's own id.

The consequences of maintaining the two partial views (V^N and V^C) is that no node holds the complete knowledge of all participating nodes or all ontology concepts. This helps applicability in a large scale setting. The gossip protocol is completely characterised by the following parameters:

- F_c: The concept fanout specifies the number of concepts a source node includes in a gossip transmission,
- F_n: The node fanout specifies the number of target nodes a gossip message is sent to,
- $age_{\text{threshold}}$: Specifies the number of times a node transmits a received concept before the concept is removed from V^C. For convenience, we define function $age(c)$ that takes concept c as input and returns its age.
- ttl_{gossip}: This value is assigned by each source node to any concepts from V^O that are selected for transmission. It specifies the number of hops that c will traverse before being discarded when $c.ttl$ reaches zero.

2.2 Gossip Protocol

We describe here a gossip protocol that transmits random subsets of concepts between the participating mobile nodes. The protocol is executed in every node and is described in terms of the actions taken during the reception and transmission of a gossip message.

On reception, a node executes the algorithm described in Listing 1. The algorithm first matches the set of concepts included in the transmission against its stored concepts. This is shown in line 3. Subsequent to matching, lines 4 – 6 show that if a concept is not found in either the concept or the ontology view and the concept's ttl is greater to one, it will be stored at the receiver's concept view.

Each gossip reception results in progressing the system's shared semantic knowledge in an incremental fashion. Through an ontology matching algorithm, described in section 3.1, each message has the potential to create new associations between concepts from different ontologies. A concept view size that is bound by the protocol and a fixed concept fanout ensure that nodes are not overwhelmed with matching large ontologies. Although the redundancy that is inherent in gossip protocols can seem excessive, it is this very feature that allows progressive matching through concept transmission and scalable discovery through concept replication. A gossip approach also avoids flooding, which in multi-hop networks can significantly increase traffic as the network size grows.

Listing 1. Gossip Reception

1: On reception of $gossip$ at node j
2: **for all** $c \in gossip.concepts$ **do**
3: *Execute ontology matching algorithm between c and* $V_j^O \bigcup V_j^C$
4: **if** $c \notin V_j^O \wedge c \notin V_j^C \wedge c.ttl > 1$ **then**
5: $c.ttl \leftarrow c.ttl - 1$
6: $V_j^C = V_j^C \bigcup \{c\}$
7: **end if**
8: **end for**

Listing 2. Gossip Transmission

1: Every t ms at node j
2: *Choose* $X = \{c_{j1}, \ldots, c_{jl}\}$ *random concepts from* $V_j^O \bigcup V_j^C$ *, with* $l = F_c$
3: **for all** $c \in X$ **do**
4: **if** $c \in V_j^O$ **then**
5: $c.ttl \leftarrow ttl_{\text{gossip}}$
6: **end if**
7: **if** $c \in V_j^C \wedge age(c) = age_{\text{threshold}}$ **then**
8: $V_j^C \leftarrow V_j^C - \{c\}$
9: **else if** $c \in V_j^C$ **then**
10: $c.age \leftarrow age(c) + 1$
11: **end if**
12: **end for**
13: $gossip.concepts \leftarrow X$
14: *Choose* $Y = \{n_1, \ldots, n_{F_n}\}$ *random nodes from* V_j^N
15: **for all** $r \in Y$ **do**
16: $\boldsymbol{send}(r, gossip)$
17: **end for**

Listing 2 describes the transmission of a gossip message. In each timeout, a node selects F_c concepts at random from the union of the ontology and concept views. Any item from the concept view that has been transmitted $age_{threshold}$ times is removed from that view. This selection algorithm is simple and exhibits a desirable adaptive behaviour. During the initial stages of gossip transmission, a node's priority is to disseminate its own ontology so that its semantic information is diffused throughout the network. Since the concept view contains few elements during the initial rounds, concepts from a node's ontology have a higher probability of being selected for transmission. A more in-depth description and analysis of the gossip protocol together with experimental results can be found in [6]. It is omitted here for space considerations.

With the dissemination of concepts by the gossip protocol, we can now discover services based on their semantic description. Concept based service discovery has been used previously in the context of P2P networks by [7] and [8]. It corresponds to discovery of services not by mere syntax but by their semantic properties. In the distributed setting that we consider here, concept discovery can provide an ideal method to identify relevant services.

3 Semantic Service Description

We have chosen the Resouce Description Framework Schema (RDFS) [9] to model ontologies and to represent service description and queries. RDFS, although limited in its modelling constructs, can provide adequate expressiveness for our demo prototype and the scenario types we consider. There is currently no service standard in RDFS, so we have defined a minimal service specification based on the service profile of OWL-S. An example of an RDFS-based service instance is shown in Fig. 1a.

In the proposed model it is concepts rather than services that are advertised and discovered. This provides the required flexibility to enable the progressive ontology matching and the concept-based discovery of services. However, the decomposition of ontologies into concepts and the distribution of concepts across the network imposes certain requirements on the concept's syntax. Ontology languages like RDFS and OWL were not designed for this task so the syntax of advertised concepts had to be augmented. We call this syntax the *network representation* of a concept to distinguish it from the normal representation a concept has in an ontology. Figure 1b shows an instance of a concept network representation. This advertised syntax needs to contain enough information to satisfy two distinct requirements:

1. *Discovery of service providers*. A first step to service discovery is the identification of nodes with ontologies compatible to the ontology of the node that initiated the discovery query. To achieve this we use the following properties. First, the gossip protocol enables the dissemination of concepts to any of the participating nodes. Second, as we show in Section 4, a discovery query can be formulated as a set of concepts. If we now embed the source node identifier in each concept network representation, we can facilitate the discovery of service provider nodes.
2. *Pair-wise concept matching*. Properties in most ontology languages (e.g. object or datatype properties in OWL) are first-class entities and are usually defined outside the scope of concepts. For the semantic interpretation of discovery queries we rely on concept matching and specifically on the name and property correspondence between concepts from different ontologies. By embedding concept properties in the network representation we can facilitate matching in any participating node. Section 3.1 elaborates on the issue.

3.1 Ontology Matching

Scarcity of computational resources and transient communication necessitates a lightweight and practical approach for matching heterogeneous ontologies. Syntactical matching is more appropriate because it requires less resources. Semantic matching on the other hand can produce more accurate integration, but requires complex inferencing over the candidate ontologies. For the initial implementation described in this paper we have used an algorithm similar to *intermediate matching* of H-Match [10]. We assume that all participating nodes share the same matching algorithm. Each node records a match between two concepts when their respective names are syntactically equal and each of their properties match in type and name. We note however that details of ontology matching are outside the scope of this work. What is of interest is the utilisation of the matching relationship after it has been established.

```
<rdf:RDF xml:lang="en"
<om:Service rdf:ID="aService">
  <om:hasInput rdf:resource="#ConceptA"/>
  ...
  <om:hasOutput rdf:resource="#ConceptB"/>
  ...
</om:Service>
</rdf:RDF>
```

a. A service instance in XML.

```
<rdfs:Class rdf:ID="C">
  <om:source rdf:resource="192.168.1.2"
  <om:isSubClassOf rdf:resource="#C"/>
  ...
  <om:hasProperty rdf:resource="P1"/>
  <om:hasProperty rdf:resource"P2"/>
  ...
</rdfs:Class>
```

b. A concept's network representation in XML.

Fig. 1. Representation of a service and a concept's advertised syntax

We define the concept matching relationship as a transitive and symmetric relation. For example, if c_i, c_j and c_k represent concepts in V_i^O, V_j^O, V_k^O; $\mathbf{M}$ the matching relation, and a match exists between c_i, c_j and also between c_j, c_k, the following matches are inferred:

1. $c_i \mathbf{M} c_j \Leftrightarrow c_j \mathbf{M} c_i$,
2. $c_j \mathbf{M} c_k \Leftrightarrow c_k \mathbf{M} c_j$
3. $c_i \mathbf{M} c_j \wedge c_j \mathbf{M} c_k \Rightarrow c_i \mathbf{M} c_k$.

The exact semantics of the matching relationship depend on the strength of the matching mechanism. Here, we have reduced the scope of matching to an equivalence relationship that is similar to the `owl:equivalentClass` property in OWL. Other relationship types are also possible, for example `kindOf` or `partOf` relationships.

There is a dependency between the network representation of concepts and the accuracy of the matching algorithm. If the network representation of a concept contains only its name, it is difficult for any algorithm to produce an accurate match between concepts. In the current prototype, the concept network representation includes any properties that have the specific concept as their domain concept in addition to properties inherited through the RDFS `subClassOf` and `subPropertyOf` relations.

4 Discovery and Matchmaking of Services in a Distributed Environment

In centralised architectures like UDDI or where broadcast facilities are available (e.g., LANs), mechanisms for service discovery can use existing infrastructure. A query to a well known URL or a broadcast request can return any available services that match certain criteria. Neither of these facilities are available in MANETs.

Our proposed mechanism for service discovery is distributed and is overlayed on top of the gossip protocol described previously. The discovery process has two phases: 1) the identification of candidate nodes with compatible ontologies and 2) the redirection of the discovery query to the candidate nodes for the final service matchmaking. During the first phase, a random walk mechanism aggregates the addresses of nodes that have ontologies with concepts that match the concepts in the discovery query. These are addresses of potential service providers since matching relationships indicate partially compatible ontologies. Depending on the matching progress and the number of hops before the discovery query expires, it is expected that candidate nodes will be identified

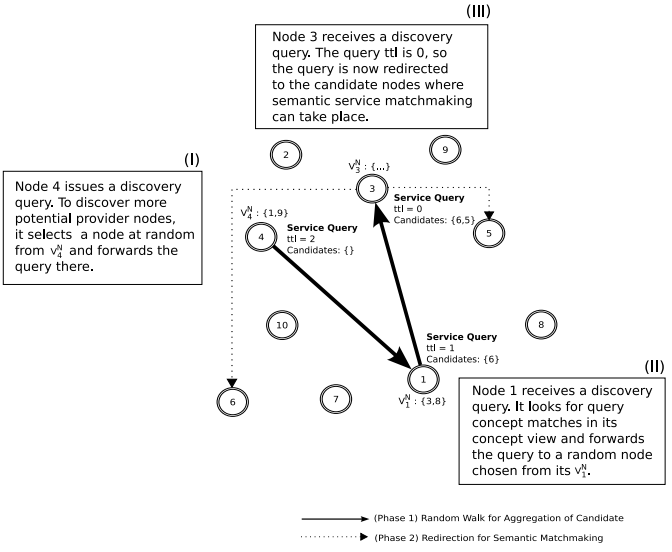

Fig. 2. The two phase semantic service discovery

with high probability. In the second phase, the query is redirected to the discovered nodes where semantic matchmaking of available services can take place. Figure 2 illustrates the process.

4.1 Aggregation of Candidate Nodes

We introduce the following definitions:

- We call a node that initiates a discovery query, the query's *source* node.
- We formulate a service query as a set of `In` and `Out` concepts. We represent such a query as:
$$Q = \{(c, T), \ldots, |c \in V^O, T = \{In, Out\}\},$$
where T is a parameter type and c a *query concept* from a node's ontology view.
- We define the *semantic context* of a concept as its super and sub-concepts.
- As the semantic context of a concept can be the complete ontology, we bound the context with a parameter τ.
- If $super_\tau(c)$ and $sub_\tau(c)$ give the set of super and sub-concepts for concept c, then we represent the semantic context of all query concepts in Q using the set:
$$H_\tau = \{super_\tau(c) \cup sub_\tau(c) | \forall c \in Q\}$$

The first step to service discovery is to identify nodes with concepts that match all the concepts in a query. A simple mechanism would identify candidate nodes by examining only the source's V^O and would immediately redirect the query to the identified nodes. This is straightforward since each concept embeds predicates for any matching concepts and their corresponding source node identifiers.

There are two problems with this approach however. First, progressive matching can take time to terminate, so concepts in a node's ontology might contain only partial matches. This can result in discovery queries with fewer hits. To increase the probability of finding candidate nodes the query is forwarded to a small subset of nodes using a random walk.

Second, semantic services provide a more flexible discovery mechanism because of subsumption-based discovery. This makes provided services described by concepts that are subsumed or subsume concepts in a query still valid. We look for matches not only for the query concepts, but also for the concepts that constitute the semantic context of the query concepts. The rationale is that if the parent of a query concept has a match, while the actual query concept has no matches, a provided service can still be compatible when defined against the concept matched by the parent. In other words, a candidate node is one that has an ontology with concepts that match *any* concept from the semantic context of *all* query concepts.

The algorithm for discovery is specified in Listing 3. We first formulate a service query by selecting a number of query concepts from a node's V^O. Apart from Q and H, a discovery query requires a third set recording the results of the random walk protocol. Note that Q and H need not contain the complete concept representation. If the qualified name of a concept includes the node's address, e.g., a URI of the form `node://<node address>/<concept name>`, that would be enough to uniquely identify concepts and avoid name clashes.

Matches are stored in $R = \{(c, \{id_1, \ldots, id_n\}), \ldots, |c \in Q \cup H\}$, where id is the identifier of a node that has an ontology with a concept matching c. To simplify our description we define a function $f_S(c)$ where c is a concept with $c \in S$ and S can be either of the two views, i.e., V^O, V^C or R. This function returns the set of node identifiers of all matched concepts currently embedded in c.

After formulating the discovery query, line 2 shows a first set of matches being recorded at the source node. In a situation where matching has been completed between all ontologies, this first set would represent all possible matches. We take the general case however and assume that matching is still ongoing (i.e., gossip protocol continues to execute). The query is then forwarded to a small number of nodes to increase the probability that all potential matches are identified. These nodes are selected using a random walk. To avoid visiting all nodes, we bound the random walk with a ttl_{query} parameter. This parameter can be set independently by each source and is necessary because the distributed nature of matching makes it hard to devise adaptive termination criteria (e.g., terminate the query when all matches are found).

Until the query's ttl reaches zero, each receiving node will augment R with any extra information it might contain and will forward the query to another random node selected from V^N. To avoid visiting a node multiple times the query records the id of each node in the random walk.

For clarity, in line 6, we omit the extra step of recording node ids by transitively following matched relationships. The condition shown, is that if the receiving node contains new matches for the query concepts, R is updated with the new matches. Line 11 is the condition to identify the set of nodes that have concepts compatible with all the query concepts or their semantic context.

Listing 3. Discovery of Service Provider Nodes

$discovery \leftarrow \{Q, H, R\}$
At source node: $R \leftarrow f_{V^O}(c), \forall c \in Q \cup H$
$discovery.ttl \leftarrow ttl_{\text{query}}$
At each node receiving discovery during the random walk:
for all $c \in Q \cup H$ **do**
  if $c \in V^C$ and $f_R(c) \subset f_{V^C}(c)$ **then**
    $R \leftarrow f_R(c) \cup f_{V^C}(c)$
  end if
end for
if $discovery.ttl = 0$ **then**
  $D = \bigcap_{c \in Q}(f_R(c) \cup \forall x \in sub_\tau(c).f_R(x) \cup \forall x \in super_\tau(c).f_R(x))$
  $\forall i \in D$: *redirect query to node* i
else
  Choose a random node id r *from* V^N
  $discovery.ttl \leftarrow discovery.ttl - 1$
  Forward query to r
end if

5 Evaluation

We have implemented the gossip protocol and the first phase of the discovery protocol. For evaluation purposes we simulated networks of 20, 40 and 60 nodes. We are interested in the evaluation of the random walk mechanism, so to simplify the simulations we used complete node views. Each node now contains the complete set of participating nodes, rather than a partial one. This simplification does not alter the correct functioning of the algorithm.

Each node maintains its own unique ontology composed of 10 concepts, i.e., $|V^O| = 10$. These are test ontologies in RDFS that are generated automatically for evaluation purposes and have no semantics. Matching between these generated ontologies is guaranteed by randomly specifying predefined matching relationships. In these predefined relationships, each concept can select another concept with a certain probability and can also be selected by other concepts provided they are not from the same ontology. Predefined matching is only an indication for the real matching relationship. Matching still materialises only when two concepts are actually compared.

We want to investigate the scalability properties of the system as the number of nodes and the number of concepts increase. We also want to investigate whether the number of concepts in $Q \cup H$ have an impact on discovery. To this effect, we have conducted repeated trials with varying network and query sizes. The following parameters held constant values: $F_c = 4, F_n = 2, ttl_{\text{gossip}} = 2, age_{\text{threshold}} = 2$.

Each of the 3 experimental setups corresponding to the different network sizes were run 20 times. In each setup, different discovery queries are run at certain system rounds. A system round is counted after every node transmits a gossip message. Every 5 rounds, the following process is repeated 30 times. A node is selected at random and a discovery

query is initiated from that node. Query concepts are also randomly selected from each source node. We issue two types of queries:

1. To assess discovery as the network size grows, we fix the query concepts to 2 and the value of ttl_{query} to 3.
2. For the evaluation of query sizes, we initiate successive queries with an increasing number of concepts that range from 1 – 4 and keep the ttl_{query} value constant at 2.

Overall, 150 queries are initiated every five rounds with each experimental setup lasting for 200 rounds.

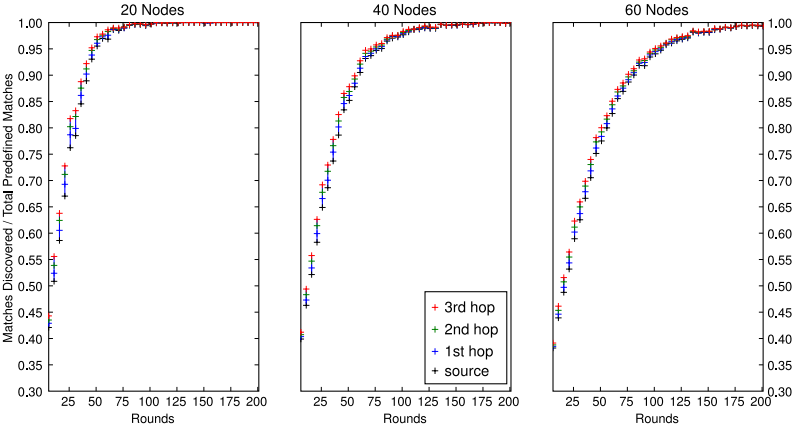

Fig. 3. Total discovery ratio vs. rounds

Figure 3 shows the mean discovery ratio between the matches discovered during the random walk and the *total* predefined matches for the query concepts. For each query and in each hop we divide the number given by $|f_R(c)|, \forall c \in Q \cup H$ against the number of predefined matches for each c. *This ratio indicates the degree to which all predefined matches have been discovered*. Any query concepts not belonging to a matching relationship are excluded from the calculation. Crosses represent the hops in the random walk. In each vertical line, the lowest cross represents the discovery ratio at the source node.

We observe that the difference in the discovery ratio between hops increases, albeit by a small percentage, as rounds progress. The overall ratio though increases exponentially with the number of rounds, until it reaches 1. This shows the strong dependency between discovery and matching. During the initial rounds, matching associations have not yet being established, so the discovery ratio is low. As more concepts are progressively disseminated across nodes, the network augments its shared knowledge.

Figure 4 shows the mean discovery ratio between the matches discovered during the random walk and the *network* matches. This ratio is calculated by dividing the number given by $|f_R(c)|$ in each hop against the number of matches that exist for concept c across the network. In other words, if c is replicated across k nodes, the network matches

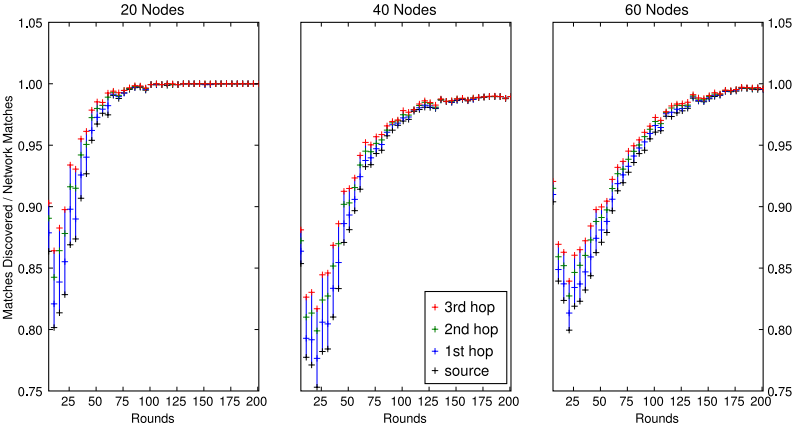

Fig. 4. Network discovery ratio vs. rounds

are calculated as: $|\bigcup_{i \in k} f_{V_i^C}(c)|$. This is an alternative measure for the performance of the random walk since it reflects the current state of ontology matching rather than the ideal state. In that respect we expect this ratio to be high in the beginning when matching associations are low, drop as more matching associations become available in the network and finally stabilise close to 1. We observe this behaviour in all cases. It is also possible for this ratio to exceed 1, as transitive matching means that a query might record more matches than what is found in the replicated query concepts.

Finally, Fig. 5 depicts the discovery ratio for queries with a varying number of query concepts. In this experiment, the discovery ratio is the value recorded on the last hop. There does not appear to be a significant difference in discovery when increasing the number of concepts. This is a good result indicating that including the semantic context

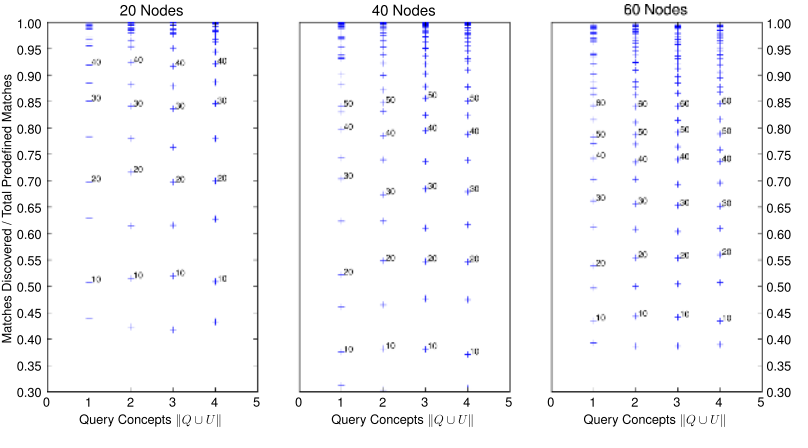

Fig. 5. Total discovery ratio vs. query concepts

of query concepts or expressing complex service queries, both of which may contain a large number of concepts, have no impact on the discovery of provider nodes. One can see more clearly in this figure the discovery latency as the number of ontologies increase. After 40 rounds, the discovery ratio is approximately 90%, 80% and 75% for 20, 40 and 60 nodes correspondingly.

If we measure scalability in terms of the discovery ratio against an increasing network size, we observe that with a bounded number of hops (ttl_{query}) the model described here eventually provides a high discovery ratio, *conditional* on the progress of ontology matching. A fundamental trade-off in the proposed model is between a discovery query that can return partial results in few hops even with a large number of provider nodes and the latency of complete results that the progressive matching approach entails. Although the effectiveness of the random walk is lower than expected, it can still increase the discovery ratio but its use should be weighted against the potential cost of extra routing traffic.

6 Related Work

The work presented here stands at the intersection of decentralised mechanisms for data dissemination and semantic services with an emphasis on scale and autonomy of interaction. Initial research in service discovery for MANET environments was mainly focused on distributed discovery protocols, e.g., [11]. This however assumed strict assumptions on service names and interfaces so that services could interoperate. Subsequent work, such as GSD [1] developed a service framework based on the semantic description of services. However, it was based upon the implicit assumption that nodes maintain a common global ontology. Current research is beginning to accept that in open distributed systems knowledge will be decentralised.

The assumption of heterogeneous domain ontologies for semantic services follows closely the evolution of semantic P2P networks. Networks such as EDUTELLA [7] assume that not only data is distributed but metadata descriptions are also decentralised and not uniform. Since in P2P networks a broader set of assumptions can be made about resource availability and peer failure rates more sophisticated techniques for ontology matching are feasible.

7 Conclusion and Future Work

We have presented a novel model to support semantic service discovery in MANETs given the assumptions of autonomous mobile nodes and heterogeneous ontologies. The model supports the progressive matching of ontologies and the decentralised discovery of semantic service provider nodes. Discovery is facilitated by a random walk mechanism that uses concept discovery to find matching concepts. As future work we plan to study the network overhead imposed by the gossip and the random walk protocol and generalise the specification of discovery queries to include Description Logics derived languages.

References

1. Chakraborty, D., Joshi, A., Finin, T., Yesha, Y.: Gsd: A Novel Group Based Service Discovery Protocol for MANETs. In: Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN'02), IEEE Press (2002)
2. Kozat, U.C., Tassiulas, L.: Network layer support for service discovery in mobile ad hoc networks. In: IEEE INFOCOM. Volume 22. (2003) 1965–1975
3. McGuinness, D.L., van Harmelen, F.: Owl web ontology language overview (2004) W3C Recommendation.
4. Aberer, K., Cudré-Mauroux, P., Ouksel, A.M., Catarci, T., Hacid, M.S., Illarramendi, A., Kashyap, V., Mecella, M., Mena, E., Neuhold, E.J., De Troyer, O., Risse, T., Scannapieco, M., Saltor, F., de Santis, L., Spaccapietra, S., Staab, S., Studer, R.: Emergent Semantics Principles and Issues. In: DASFAA 2004. (2004) 25–38
5. Aberer, K., Cudré-Mauroux, P., Hauswirth, M.: A framework for semantic gossiping. SIGMOD Rec. **31** (2002) 48–53
6. Nedos, A., Singh, K., Cunningham, R., Clarke, S.: A Gossip Protocol to Support Service Discovery with Heterogeneous Ontologies in manets. Technical Report TCD-CS-2006-34, Distributed Systems Group, Computer Science Department, Trinity College Dublin (2006)
7. Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., Risch, T.: Edutella: A P2P Networking Infrastructure Based on RDF. In: Proceedings of the 11th International Conference on World Wide Web (WWW'02), New York, NY, USA, ACM Press (2002) 604–615
8. Castano, S., Ferrara, A., Montanelli, S., Pagani, E., Rossi, G.: Ontology-Addressable Contents in P2P Networks. In: Proceedings of the 1st WWW International Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID '03), Budapest, Hungary (2003)
9. Brickley, D., Guha, R.: Resource Description Framework RDF Schema Specification 1.0 (2000) W3C.
10. Castano, S., Ferrara, A., Montanelli, S.: H-match: an algorithm for dynamically matching ontologies in peer-based systems. In: SWDB. (2003) 231–250
11. Kozat, U.C., Tassiulas, L.: Service discovery in mobile ad hoc networks: an overall perspective on architectural choices and network layer support issues. Ad Hoc Networks **2** (2004) 23–44