A Gossip Protocol to Support Service Discovery with Heterogeneous Ontologies in MANETs

Andronikos Nedos, Kulpreet Singh, Raymond Cunningham, and Siobhán Clarke

Distributed Systems Group, Trinity College Dublin, Ireland

Abstract. Service discovery in mobile ad hoc networks (MANETs) is an integral part for collective application interoperability. The discovery process must cope not only with transient communication but also with an environment where autonomous mobile nodes act as both service providers and consumers. Imposing predefined service interfaces in such an unpredictable and dynamic environment is an inapppropriate assumption. A more flexible description and discovery mechanism can be provided with the use of ontologies and semantic reasoning. Assuming that services are described by heterogeneous ontologies poses many technical challenges but is more realistic than requiring a single domain ontology. In particular, a mechanism is required to match the different ontologies and make provided services available to all nodes. In this paper we present a model to support semantic service discovery in MANETs. A core part of the model is the distributed approach to ontology matching. We rely on the use of a novel gossip protocol that offers an even utilisation of physical resources across the participating nodes. This makes the model suitable for resource constraint mobile devices. We present the gossip protocol and an evaluation that shows the good scalability and discovery properties of the protocol.

1 Introduction

The composition of software applications as a set of modular services that can be advertised and discovered on demand is considered a key element in distributed application interoperability. Current service description and discovery mechanisms in fixed networks (e.g., SLP [1], Jini [2], UPnP [3], etc.) and in MANETs (e.g., [4], [5], [6]) rely on standardised interfaces to achieve the necessary consensus that makes advertising and discovery possible. While this is usually sufficient for environments that are not very dynamic or are centrally administered, it poses a serious problem when one requires uncoordinated interaction in distributed environments. This is a challenge that is faced currently by open systems such as peer to peer and mobile ad hoc networks.

In MANETs the challenge in service discovery lies in engineering efficient and scalable discovery protocols and also in using service descriptions that are sufficiently expressive. While interface or attribute-based discovery queries can be fast and effective they are not very expressive and require a degree of standardisation to be meaningful. It has been argued in [7] and [8] that ontologies can serve as a flexible service description vocabulary and as an appropriate mechanism for the discovery and the semantic matchmaking of services. Work in the area of semantic services for MANETs has been limited with the exception of [4] where every node is required to specify services in a common domain ontology.

However, given the opportunistic and unpredictable interaction patterns in MANETs it would seem inappropriate to assume that a single ontology is available in every mobile node. It is more realistic to apply the same decentralised properties that permeate communication to the semantic layer. This semantic decentralisation corresponds to heterogeneous ontologies that are developed autonomously and are maintained at individual nodes.

Current research trends in data semantics and ontologies have started to expand the basic assumption of a common unified schema that is known by all participants [9]. It is now assumed that knowledge will be distributed and even metadata will be developed autonomously. Since a shared understanding is still required for meaningful semantic interpretation, an ontology matching process is needed to produce a shared ontology from different ones. However, even in settings where computational resources are plentiful and the network topology is relatively stable, availability of network resources (e.g., WordNet [10]) or expert intervention might still be required to resolve semantic mismatches. In ad hoc networks the problem is compounded by limited resources, a requirement for fully automated operation and failure-prone communication.

We demonstrate one aspect of this problem, namely the processing cost of ontology matching, in Table 1.

	Properties					
Concepts	*0*	*1*	*2*	*3*	*4*	*5*
10	0.2410	0.4455	0.7162	1.0374	1.3829	1.8425
20	0.5557	1.2816	2.2467	3.3018	4.5969	6.0430
30	1.0649	2.5754	4.5777	6.9349	9.6372	12.8218
40	1.6876	4.2554	7.7056	11.5536	16.2711	21.9438

Table 1. Time in seconds required for the pairwise ontology matching in a Compaq IPAQ H3870 running linux. Ontologies are described in XML/RDF(S) and the matching algorithm is implemented in python using the rdflib library. The matching algorithm is a simple string matching of concept names, properties and their types.

It displays the overhead in time for performing simple syntactic matching between various ontologies. These test ontologies are of different sizes and are composed of concepts containing a variable number of properties. The processing time is an indication of the expected overhead even when simple string matching algorithms are used to match different ontologies. The table shows that even simple matching between average sized ontologies can create excessive load. An alternative approach to alleviate some of the performance bottlenecks associated with centralised matching is to match ontologies in a partial and progressive fashion. The inherent distribution properties of partial matching also make it an ideal candidate for the environment we consider here.

1.1 Problem Statement and Contribution

The problem investigated here is expressed as following: *given autonomous nodes in a MANET that can be both providers and consumers of services and services that can be described by different domain ontologies, there is a need for network-wide service access to and from the autonomous nodes.*

The work presented here describes an efficient and scalable model as a solution to the above problem. We assume that ontologies are composed of concepts and services are specified using these concepts. We have devised a progressive ontology matching mechanism which can also facilitate concept-based service discovery. At the core of the model is a partial view that nodes maintain over all available concepts and a gossip protocol that disseminates randomised subsets of concepts. The execution of a matching algorithm in each node enables an even utilisation of resources, while the inherent replication stemming from the partial views aids the process of semantic service discovery. The intuition behind this model is to augment the environment's shared knowledge by progressively matching all concepts and replicate part of this knowledge in every node.

The paper is organised as follows. In section 2 we provide an introduction to the model and how it supports the distributed discovery of semantic services. Section 3 describes the details of the gossip based protocol and the ontology matching algorithm. Section 4 presents an implementation with ns-2 and the evaluation results while we conclude with the state of the art and final remarks in sections 5 and 6.

2 A Model for Semantic Service Discovery in MANETs

In an Internet environment a host of semantic service standards (e.g., WSDL-S, OWL-S, WSMO) and protocols (e.g., UDDI, SOAP, etc.) are available to enhance interoperability and enable service interaction. In addition, many different network topologies (e.g., centralised UDDI directories, decentralised P2P) can be used depending on the required discovery properties. Two central assumptions about this environment are the *persistence of references* and the *availability of resources*. Even with a decentralised network topology, a modest assumption about longevity of ontology references can be made. Clients need only know a URI reference to obtain an ontology. On the other hand, resource availability means that sophisticated schemes can be devised for matching ontologies and the discovery of services.

The above assumptions do not hold when mobile nodes form an ad hoc network. The transient nature of communication means that even the weakest assumptions about the existence of ontology references cannot be guaranteed. Only ontologies maintained by currently connected nodes can be referenced and only for the period that a node remains connected. On the other hand, physical resources, (e.g battery life, CPU power, bandwidth) are scarce and require consideration when trying to exchange and match different ontologies.

2.1 Concept Discovery

We envisage the following generic process for the semantic discovery of services:

1. a service request is formulated as a concept-based query,
2. concepts from other nodes that match the queried concepts are discovered,
3. the matching concepts provide a list of candidate nodes that host compatible ontologies,
4. the query can now be redirected to the candidate nodes where semantic matchmaking can take place between the required and advertised services.

This paper is focused on the first three steps and provides the fourth step for completenes. The first step reformulates the discovery of semantic services as a concept discovery problem. In turn, concept discovery depends on ontology matching and the availability of these ontologies for quering. To facilitate these dependencies we employ a single distributed mechanism for both ontology matching and concept discovery. In particular, we use concept replication for concept discovery and continuous concept dissemination for the eventual matching of the heterogeneous ontologies. To this end we employ a novel gossip protocol to exchange randomised sets of concepts between nodes. Received concepts are then stored in a buffer maintained by each node until certain conditions are met. It is instructive to envisage this buffer that contains a random subset of all available concepts as a view on all ontologies.

This view (i.e., buffer) 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.

Section 3.1 describes the details of the gossip protocol. The transmission of a fixed number of random concepts is coupled with matching by every participating node to achieve an even distribution of the processing load. Such a model is inherently distributed avoiding clustered solutions in which cluster-heads must store all available ontologies. Also, semantic inference for the matchmaking of services (step four) takes place at the provider node thereby localising the associated overhead. This approach has the following characteristics:

Provides a substrate for the distributed discovery of semantic services As in most semantic frameworks we consider concepts to constitute the basic reference units. Concepts are used for matching and also for the description and discovery of services. For matching and discovery we use a custom concept representation that augments the normal representation of concepts when defined inside an ontology. We call this the *network representation* and is defined in section 3.2. For the description of services we assume a simplified version of the OWL-S profile where a service description is composed by a sequence of input and output concepts. Service discovery can now be reduced to finding the set of concepts that match those composing the service request. Because concepts are replicated and a partial membership is available in every node, a simple random walk protocol can be used to discover matching concepts.

Scalable and unconfined service interaction when node and ontology sizes increase : Gossip protocols are frequently used to improve scalability and efficiency in peer communication. For semantic service discovery the participating nodes and the ontology sizes are key factors that can influence scalability. If we also consider the multiple interactions that are necessary to discovery, match and fetch available services, scalable discovery is even more critical. Our simulations demonstrate that concept discovery is probabilistically bound to a small number of hops even when large number of concepts are used. Additionally, it can be shown through analysis borrowed from the epidemic literature that as the number of nodes (and hence ontologies) increase, only a moderate increase in resources is required.

3 System Model

We provide here the set of assumptions and notation that we use throughout the rest of the paper. We consider an ad hoc network as composed of a set $\mathcal{N}$ of mobile nodes. A subset $\mathcal{P} = \{n_1, n_2, ..., n_m\} \subseteq \mathcal{N}$ of nodes are considered to be active participants that maintain ontologies and share services. To simplify the description of the model we assume that all nodes in $\mathcal{N}$ are initially uniformly located across a geographical area. We also assume that the initial placement of nodes in $\mathcal{P}$ is also uniform across the nodes in $\mathcal{N}$. Subsequent movement of nodes can be arbitrary and we do not pose any restrictions on the mobility model.

All nodes are addressed by a unique identifier. Nodes that are not active participants need only have the basic capability of forwarding packets. We assume that nodes communicate using a fixed-range, wireless medium (e.g., IEEE 802.11) and that a unicast protocol (e.g., AODV, OLSR, etc.) is available.

Each node $n_i \in \mathcal{P}$ maintains three different views (i.e., buffers). The *Ontology View* (V_i^O), the *Concept View* (V_i^C) and the *Node View* (V_i^N). Although we use the term view uniformly for all required buffers, we are mainly interested in the properties and performance characteristics of the concept view. A node's ontology is composed from a set of concepts and is stored at the V^O. We assume that for the lifetime of the protocol execution no additions or deletions of concepts occur in this view. Any received concepts are stored in the concept view.

The node view is comprised by a set of node identifiers. It maintains a uniform, randomised, partial and fixed-size set of node ids and is populated during a bootstrap phase. The consequences of maintaining the two partial views (V^N and N^C) is that no node holds the complete knowledge of all participating nodes or all ontology concepts. Both the concept and the node views do not allow duplicate entries. Additionally, the node view excludes the source node address while the concept view does not store concepts found in the node's ontology view.

The protocol defined below is completely characterised by the following parameters:

- F_c, the concept fanout specifies the number of concepts a source node transmits during a gossip round,
- F_n, the node fanout specifies the number of destination nodes a gossip message is sent to,
- $age(c)$, age specifies the number of times a node transmits concept c before it is removed from the concept view,
- $ttl(c)$, time to live is a parameter that specifies the number of times concept c can be forwarded before being discarded when $ttl(c)$ equals zero.

3.1 Gossip Protocol and Semantic Matching

Bootstrapping The gossip protocol consists of two phases, *reception* and *transmission*. A node must first populate its node view with a partial membership of other participants, before being allowed to gossip. A simple bootstrapping protocol helps achieve this. The bootstrapping protocol is based on a simple *expanding ring search* and is used when a node enters an ad hoc network or after a node fails. An initial request is first flooded through the network with each receiving node sending a reply back to the source node with some probability. During this initial exchange, both source and receiving nodes populate their node views with the node ids found on these initial messages. Using definitions borrowed from the gossip literature [11], the bootstrap protocol is a *pull* gossip while the normal protocol execution is a *push* gossip. A node enters normal protocol execution when it receives enough responses so that its node view size exceeds a threshold.

1 Gossip reception

```
On reception of $gossip$ at node $j$
for all $c \in gossip.concepts$ do
    Execute ontology matching algorithm between c and $V_j^O \bigcup V_j^C$
    if $c \notin V_j^O \wedge c \notin V_j^C \wedge ttl(c) <$ ttl-threshold then
        $V_j^C = V_j^C \bigcup \{c\}$
    end if
end for
$V_j^N = V_j^N \bigcup gossip.nodes$
if $|V_j^N|$ exceeds threshold then
    Prune $V_j^N$ by removing random node ids
end if
```

Protocol On reception of a gossip message, a node executes the algorithm shown in Algorithm 1. The algorithm takes the set of concepts included in the gossip message and matches them against its stored concepts. Subsequent to matching a concept is stored in the receiver's concept view if it is not already in the concept or ontology views and the concept's ttl has not reached zero. Each new concept reception can be seen as "infecting" the receiving node. At any point the nodes that are infected by a single concept are the nodes that maintain the said concept in their concept views. It follows that nodes can be infected by multiple concepts and reinfected by the same concept. This is obvious since multiple distinct concepts are maintained in each concept view and each node can receive the same concept multiple times.

During reception there is a significant tradeoff between the processing overhead associated with the pairwise matching of received and stored concepts and the desired progress in ontology matching. If we choose to match incoming concepts with every item in the ontology and concept views, the processing overhead can be quite high. This however increases the rate of ontology matching as received concepts are matched against a larger set of concepts in the two views. An alternative would be to match incoming concepts against concepts found only in V^O which has significantly lower performance impact but which also results in a slower rate of ontology matching. Another alternative would be to match incoming concepts against a fixed but randomised subset of concepts from both views. We offer some insight into the rate of ontology matching in the evaluation section.

Each gossip reception progresses the system's shared semantic knowledge in an incremental fashion. Through the ontology matching algorithm each message can result in new associations between concepts from different ontologies. A benefit of this model is the imposed bound on the concept view. This bound is probabilistic rather than parametric and is not set explicitly but rather depends on the protocol parameters and the system ontology size. The bounded view size and the fixed concept fanout ensures that nodes are not overwhelmed with matching large ontologies as shown in Table 1. Although the replication that is inherent in gossip protocols can seem excessive, it is this very feature that allows progressive matching and a probabilistic bound on discovery.

2 Gossip transmission

```
Every $t$ ms at node $j$
Choose $\{c_1, c_2, \ldots, c_{F_c}\}$ random concepts from $V_j^O \bigcup V_j^C$
for all $c \in \{c_1, c_2, \ldots, c_{F_c}\}$ do
  if $age(c) ==$ age-threshold $\wedge\ c \in V_j^C$ then
    Remove c from $V_j^C$
  else
    $age(c) + +$
  end if
end for
$gossip.concepts \leftarrow \{c_1, \ldots, c_{F_c}\}$
Choose a random node id $r$ from $V_j^N$
$gossip.nodes \leftarrow \{j, r\}$
Choose $\{n_1, \ldots, n_{F_n}\}$ random nodes from $V_j^N \bigcap r$
for all $i \in \{n_1, \ldots, n_{F_n}\}$ do
  $\boldsymbol{send}(i, gossip)$
end for
```

Algorithm 2 shows the transmission of a gossip message. After a timeout, each 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[1]. Subsequently, as the concept view increases in size it fluctuates around a probabilistic bound. Progressively, the inclusion probability of an ontology view concept becomes lower as the size of the concept view increases.

3.2 Ontology Matching in MANETs

Many ontology matching techniques have been proposed in the literature [12, 13]. However, specific properties of the MANET environment and our model restrict the choices of a suitable matching algorithm. For example,

1. *scarcity of physical resources* – prevents the use of matching techniques that require a lot of computational resources,
2. *transient communication* – hampers time-consuming and elaborate matching techniques strengthening the need for a progressive matching approach,
3. *randomisation* – detaches concepts from their context. The gossip protocol selects concepts for transmission independent of their adjacent concepts. As such the context of a concept, comprised by its adjacent concepts in the ontology graph, is often not transmitted. This prevents the usage of matching techniques like the deep and intensive model in H-Match [13] where the context of a concept is used to draw more accurate results.

The combination of the aforementioned factors necessitates a lightweight and practical approach. Syntactical matching requires less resources so it is more appropriate in this context. On the other hand, semantic matching can produce more accurate integration but requires complex inferencing over complete candidate ontologies. For the proof of concept implementation described in this paper we have used a model similar

---

[1] The inclusion probability of a concept in V^O appearing in the concept fanout is actually: $\frac{\binom{i+j}{F_c}-\binom{i}{F_c}}{\binom{i+j}{F_c}}$ where $|V^C| = i, |V^O| = j$ and $i > F_c, j > F_c$. This probability is higher when $|V^C|$ is small.

to *intermediate matching* of the H-Match algorithm. A match between two concepts is recorded when their respective names correspond and each of their properties match in type and name.

Contrary to other service discovery approaches, services per se are not advertised. It is their constituent elements, concepts in this case, that are advertised and discovered. This decomposition of individual ontologies into concepts and their distribution across the network imposes certain requirements on the concept syntax. Ontology languages like RDFS and OWL were not designed for this task so our model requires an augmented syntax for concepts that are distributed across the network.

We call this syntax the *network representation* of a concept to distinguish it from the normal representation in a local ontology. In short, the network representation aids pair-wise concept matching and discovery of service providers by including references to concept properties and source node identifiers. An instance of a concept's network representation if shown below:

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

Although matching is architecturally separate from the gossip protocol, there is a dependency between a concept's network representation and the accuracy of the matching algorithm. A concept that has only its name in the network representation will make it difficult for any matching algorithms to produce an accurate match. In the current prototype the network representation of a concept includes references not only to the set of properties this concept is in the domain of but also to the properties inherited by its super concepts.

We define the matching relationship as a transitive, symmetric relationship between two concepts. For example, if c_i, c_j and c_k are concepts, **M** the matching relationship and a match exists between c_i, c_j and also between c_j, c_k, then: $c_i M c_j \Leftrightarrow c_j M c_i$ and $c_i M c_j \wedge c_j M c_k \Rightarrow c_i M c_k$.

Fig. 1 illustrates the matching of concepts between two hypothetical ontologies. The network representation for two of the concepts is shown in each box. It contains the concept's name, its source node identifier and a list of properties and their range types. Similarity in the names and properties results in a bidirectional matching relationship between the concepts with name `#C`. A match is realised as two extra predicates in the concept definition. These predicates take as values the URI of the matched concept and the address of its source node.

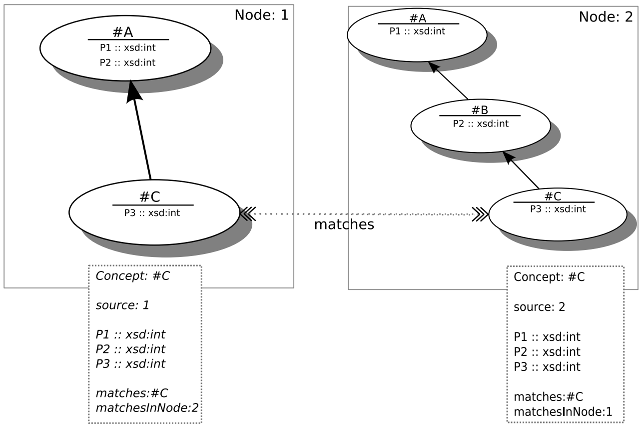

Fig. 1. Creating matching associations between concepts.

Establishing transitive, symmetric relationships in a decentralised way can easily lead to inconsistencies and matching loops. Such a scenario can arise for example when concept c_i matches concept c_j, which in turn matches concept c_i'. We rely on the usage of the matching relationship to avoid these problems.

In here, we have detailed a concept exchange protocol and an associated concept matching mechanism that provide the network support for steps one, two and three of the service discovery process as presented in section 2.1.

4 Implementation and Evaluation

We implemented the protocol in the ns-2 simulator. We used RDF(S) for describing the individual ontologies since it is more lightweight than OWL-S, albeit with weaker inference semantics. The simulation uses an area of $1500m \times 500m$ with the node sizes of $\mathcal{N}$ and $\mathcal{P}$ being 70 and 42 respectively. For the mobility model we selected a steady-state random waypoint model [14] for more accurate simulation results. Each node ontology is composed of 10 concepts, bringing the total number of concepts to 420. In practical terms an ontology with 420 terms is relatively large and uncommon. However, we specifically chose a large number of concepts to test the scalability of the model.

The purpose of the present evaluation is to demonstrate the model fitness in terms of *concept discovery* and *ontology matching latency*. We define concept discovery as a function of the number of hops required before a concept is found. We use a variation on the random walk model to find a node that contains the queried concept. Ontology matching latency is specified relative to the number of rounds required for the complete pair-wise matching between all available concepts.

We distinguish two cases of matching latency that are based on whether we match received concepts against both the V^C and V^O or just the V^O. We call the former *transient* and the latter *persistent* matching. Transient matching contains an element of redundancy. The reason for this is that any matching associations established in the concept view may be discarded. Some concepts will not be propagated due to ttl reaching zero or because of message loss. On the other hand, associations persist when received concepts are matched against the V^O. Obviously, continuous matching between each received concept and those in both views entails high processing overhead and a lot of replication. This extra overhead comes with the benefit of faster pair-wise matching as indicated by the evaluation results.

Gossip protocols tend to involve a significant number of parameters. We have chosen to demonstrate the performance of our protocol by varying the characteristic parameters of age and ttl. Since the goal of the paper is not to find the optimum parameters for the gossip protocol we have kept the node and concept fanout at constant values. A node fanout of two strikes a good balance between network traffic and matching progress. Experiments in [15] with different fanout parameters have also concluded for a small value of F_n. For the concept fanout, a value of four allows selected concepts to fit in a single gossip transmission.

Fig. 2 illustrates the distribution of concept view sizes amongst nodes for different values of ttl and age. Incrementing the value of age doubles on average the size of the view for the same value of ttl. The age parameter increases replication, hence discoverability, but requires increased storage and processing resources.

Fig. 3 depicts the infection progress. In this test, each concept is considered a single "virus" that must infect all participating nodes in order to complete the matching. We plot the percentage of nodes that have been infected by a percentage of concepts at certain round intervals. This corresponds to persistent matching with associations being established only between the received concepts and concepts in the ontology view. As shown in the figure, increasing age and ttl results in a higher proportion of concepts infecting a higher proportion of nodes in relatively fewer rounds.

Fig. 4 displays the matching latency as each received concept is matched against both the V^C and the V^O. We see that a pair-wise matching can be achieved very quickly with the age and ttl parameters having

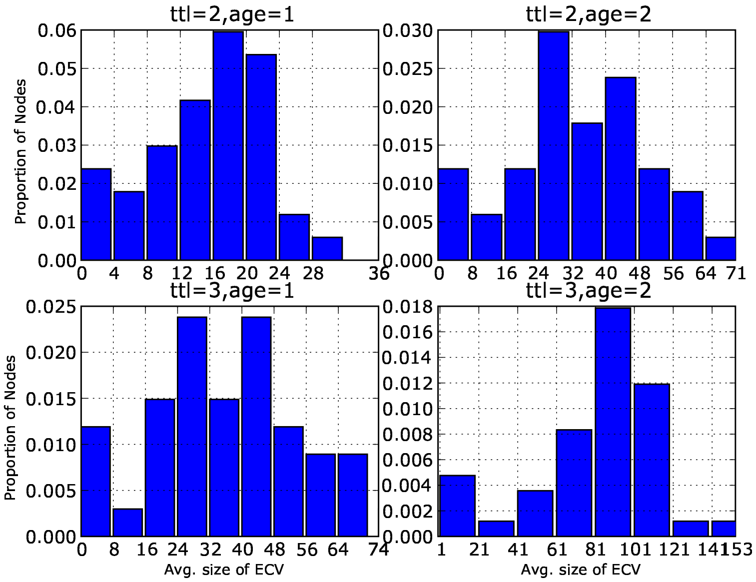

Fig. 2. The distribution of concept view sizes between nodes.

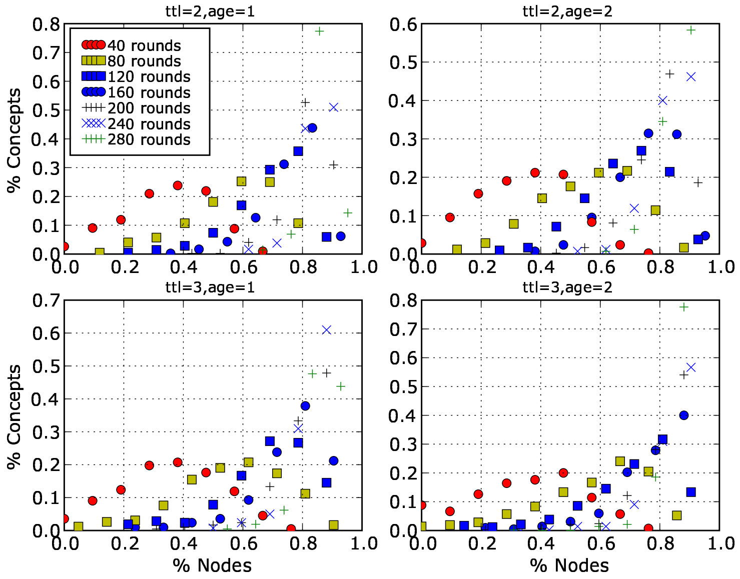

Fig. 3. Infection spread as concepts infect nodes.

minor influence. This is an overestimation however since some of the matching associations will not be utilised for reasons that were mentioned above.

Finally, Fig. 5 plots the concept discovery performance as a function of the number of hops required before a concept is found. We simulated discovery, by having a random set of nodes initiate a discovery query at regular intervals. Each discovery query is for a concept that is known to exist but it is not in the initiating node's ontology view. During discovery a node will first check its own concept view and if the

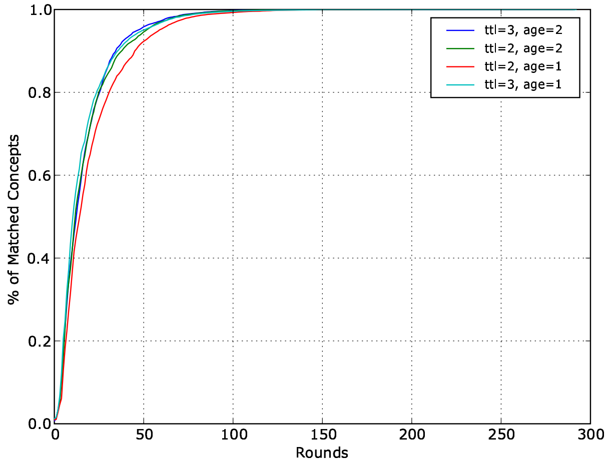

Fig. 4. Percentage of pair-wise concept matching versus rounds.

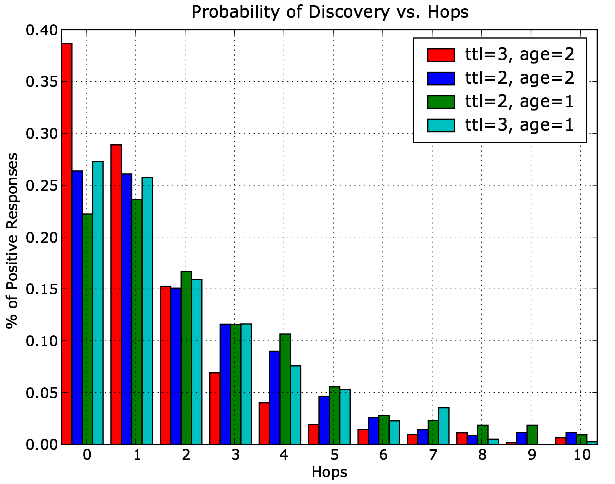

Fig. 5. Distribution of concept discovery vs. number of hops required.

concept is not found, a target node is chosen at random from its node view. Predictably, the higher the replication of concepts across the nodes the fewer hops a discovery query needs to traverse.

4.1 Discussion

The protocol presented here remains unoptimised in several areas. In terms of the overhead associated by maintaining the overlay network in each participant's node view, most of the optimisations applied in [16] can also apply here. However, since this protocol operates higher in the application stack, independence from the routing layer is a major requirement. Although we have used a reactive routing protocol (AODV) for the simulations, we believe that a proactive routing protocol like OLSR would give better performance. The use of a proactive routing protocol would reduce the need for constant route discoveries.

The protocol also requires an explicit leave operation, where a node wishing to depart multicasts a leave request to several participants chosen from his node view. Subsequently, the leave request propagates through the network with receiving nodes removing matching associations from any concept containing references to the departing node.

5 Related Work

The work presented in this paper is a synthesis of research in semantic service discovery, ontology matching and gossiping protocols. The next paragraphs describe important results and show the rationale behind some of our design decisions.

Initial research in service discovery for MANET environments included work mainly in distributed discovery protocols (e.g., [17], [18], [19], [6]). This however assumed strict assumptions on service names and interfaces so that services could interoperate. Subsequent work, such as GSD [4] 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.

Research on semantic service discovery has been primarily motivated by the rise of the Semantic Web as a platform for the uncoordinated interchange of information. Through the use of semantic inference and ontologies, it is envisioned that a richer interaction model will emerge. One of the goals for realising this vision is the semantic matching of services. Projects like [20], [21] and [22] have provided a methodology for the semantic matchmaking of services in the context of the web. In particular, [21] uses OWL-S for service description and derives a degree of matching on a discrete scale between a service request and service advertisements that is based on semantic affinity.

The decentralisation of ontologies for semantic services follows closely the evolution of semantic P2P networks. Networks such as EDUTELLA [9], 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 schema mapping techniques are feasible. A practical model for ontology mapping is presented in [13].

Finally, gossip protocols have been used in many network topologies to achieve a tradeoff between efficiency and reliability. In [23] and [16], gossip based protocols have been used to increase reliability and scalability in Internet-based and MANET environments. In [15], a gossip protocol is developed in the context of a P2P publish-subscribe system that provides a partial but fixed membership of the participating members. In our model the node view is modelled after the partial membership described in the lightweight probabilistic broadcast paper. The main difference between the gossip protocol presented here and multicast gossip protocols is the assumption of a large but finite set of data elements (concepts) that need to be pairwise matched by being delivered to all participating nodes. Multicast gossip protocols usually involve the transmission of packets until all participating nodes receive them.

6 Conclusion and Future Work

In this paper we have presented a novel model to support semantic service discovery when heterogeneous ontologies are used in a MANET environment. The model allows progressive ontology matching so that a shared emergent ontology can arise, while offering discovery of services through replication. The model's reliance on heterogeneous ontologies makes it suitable for transient and unpredictable interactions. It uses a randomised gossip protocol that allows tunable performance of concept discovery and matching latency. We are currently working on modelling the stochastic behaviour of the protocol and hope to incorporate the characteristic parameters in this analysis.

References

1. Guttman, E., Perkins, C., J.Veizades, M.Day: Service Location Protocol, Version 2. IETF. (1999) http://www.rfceditor.org/rfc/rfc2608.txt.

2. Arnold, K., Scheifler, R., Waldo, J., O'Sullivan, B., Wollrath, A., O'Sullivan, B., Wollrath, A.: Jini Specification. Addison-Wesley Longman Publishing Co., Inc. (1999)
3. Corporation, M.: Universal plug and play: Background (1999)
4. Chakraborty, D., Joshi, A., Finin, T., Yesha, Y.: Gsd: A Novel Group Based Service Discovery Protocol for MANETs. In: In 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN '02), IEEE Press (2002)
5. Helal, S., Desai, N., Verma, V., Lee, C.: Konark – a Service Discovery and Delivery Protocol for Ad-Hoc Networks. In: Proceedings of the 3rd IEEE Conference on Wireless Communication Networks (WCNC '02), IEEE (2002)
6. Kozat, U.C., Tassiulas, L.: Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In: IEEE INFOCOM. Volume 22. (2003) 1965–1975
7. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K.: OWL-S: Semantic Markup for Web Services (2004)
8. 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 invited paper.
9. 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
10. Miller, G.A.: Wordnet: a Lexical Database for English. Communications of the ACM **38**(11) (1995) 39–41
11. Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., Terry, D.: Epidemic Algorithms for Replicated Database Maintenance. In: Proceedings of the 6th annual ACM Symposium on Principles of Distributed Computing (PODC '87), New York, NY, USA, ACM Press (1987) 1–12
12. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to Map between Ontologies on the Semantic Web. In: Proceedings of the 11th International WWW Conference. (2002)
13. Castano, S., Ferrara, A., Montanelli, S.: H-match: an algorithm for dynamically matching ontologies in peer-based systems. In: SWDB. (2003) 231–250
14. PalChaudhuri, S., Boudec, J.Y.L., Vojnovic, M.: Perfect Simulations for Random Trip Mobility Models. In: ANSS '05: Proceedings of the 38th annual Symposium on Simulation, Washington, DC, USA, IEEE Computer Society (2005) 72–79
15. Eugster, P.T., Guerraoui, R., Handurukande, S.B., Kouznetsov, P., Kermarrec, A.M.: Lightweight Probabilistic Broadcast. ACM Transactions on Computer Systems **21**(4) (2003) 341–374
16. Luo, J., Eugster, P.T., Hubaux, J.P.: Probabilistic Reliable Multicast in Ad Hoc Networks. Ad Hoc Networks **2**(4) (2004) 369–386
17. Haas, Z.J., Liang, B.: Ad-hoc mobility management with randomized database groups. In: IEEE International Conference on Communications (ICC '99). Number 1 (1999) 1756–1762
18. Xue, Y., Li, B., Nahrstedt, K.: A Scalable Location Management Scheme in Mobile Ad-Hoc Networks. In: Proceedings of the 26th Annual IEEE Conference on Local Computer Networks, IEEE Computer Society (2001) 102–112
19. Li, J., Jannotti, J., De Couto, D.S.J., Karger, D.R., Morris, R.: A scalable location service for geographic ad hoc routing. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM Press (2000) 120–130
20. Trastour, D., Bartolini, C., Gonzalez-Castillo, J.: A Semantic Web Approach to Service Description for Matchmaking of Services. In: Semantic Web Working Symposium, SWWS. (2001) 447–461
21. Paolucci, M., Kawamura, T., Payne, T.R., Sycara, K.P.: Semantic Matching of Web Services Capabilities. In: Proceedings of the 1st International Semantic Web Conference on The Semantic Web ISWC '02, London, UK, Springer Verlag (2002) 333–347
22. Sycara, K., Widoff, S., Klusch, M., Lu, J.: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. Autonomous Agents and Multi-Agent Systems **5**(2) (2002) 173–203
23. Birman, K.P., Hayden, M., Oskasap, O., Xiao, Z., Budiu, M., Minsky, Y.: Bimodal Multicast. ACM Transactions on Computer Systems (TOCS) **17**(2) (1999) 41–88