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

Andronikos Nedos Kulpreet Singh Raymond Cunningham Siobhán Clarke
Trinity College Dublin
Department of Computer Science
Distributed Systems Groups
E-mail: {anedos,singhk,rcnnnghm,sclarke}@cs.tcd.ie

Abstract

Service discovery is vital in enabling interoperability of distributed service-based applications. In mobile ad hoc networks (MANETs), discovery must cope not only with transient communication but also with an environment in which mobile nodes are autonomous and connectivity is opportunistic. The use of syntactic service interfaces in MANETs requires a priori agreement on interface names, limiting node autonomy and the range of provided services. A more flexible discovery mechanism can be provided with the use of ontologies. Since the use of a single ontology to describe all services also requires a priori agreement on a common semantic representation, a more realistic assumption is the use of heterogeneous ontologies. However, this assumption poses many technical challenges by requiring a mechanism to match the different ontologies and make provided services available to all nodes. This paper presents a model to support such semantic service discovery in MANETs. A core part of the model is the distributed approach to ontology matching and service discovery. We rely on the use of a novel gossip protocol to randomly disseminate ontology concepts, thereby creating a semantic overlay network. We present the gossip protocol and an evaluation demonstrating its scalability and discovery properties.

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 the interoperability of distributed applications. Current service description and discovery mechanisms in fixed networks e.g., SLP [11], Jini [2], UPnP [17], etc., and in MANETs e.g., [4], [10], [12], rely on standardised syntactic 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 opportunistic interaction in distributed environments. This is a challenge that is currently faced 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, while also using service descriptions that are sufficiently expressive. Although syntactic interfaces provide a simple and practical description, they are less expressive and require a degree of standardisation to be meaningful. It has been argued in [9] and [1] that ontologies can serve as a flexible service description vocabulary and as an appropriate mechanism for the discovery and semantic matchmaking of services. Work in the area of semantic services for MANETs has been limited, with the exception of the Group-based Service Discovery Protocol (GSD) [4], where every node is required to specify services in a common domain ontology.

However, given the opportunistic networking aspect of MANETs, it would seem inappropriate to assume that a single ontology will be used by 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 [21]. 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 single shared ontology from heterogeneous ontologies. However, even in settings where computational resources are plentiful and the network topology is relatively stable, availability of network resources e.g., WordNet [18], or expert intervention may 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.

It displays the overhead in terms of time taken for performing syntactic matching between two ontologies. These test ontologies are of different sizes and are composed of concepts containing a variable number of properties. Concepts model similar classes of objects and in the domain of Knowledge Representation (KR) concepts are composed in hierarchies representing domains of interest. The processing time is an indication of the expected overhead when simple string

Third IEEE International Conference on
Wireless and Mobile Computing, Networking and Communications (WiMob 2007)
0-7695-2889-9/07 $25.00 © 2007 IEEE

Concepts	Properties 0	1	2	3
10	0.2410	0.4455	0.7162	1.0374
20	0.5557	1.2816	2.2467	3.3018
30	1.0649	2.5754	4.5777	6.9349
40	1.6876	4.2554	7.7056	11.5536

Table 1. Time in seconds required for the pair-wise matching between concepts of two ontologies in a Compaq IPAQ H3870 running linux. The matching algorithm is a simple string matching of concept names, properties and their types.

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 that alleviates some of the performance bottlenecks associated with centralised matching is the matching of ontologies in a *partial* and *progressive* fashion. The inherent distribution properties of partial matching also make it an ideal candidate for MANETs.

1.1 Problem Statement and Contribution

The problem investigated in this paper is expressed as follows: *MANET services that are developed independently and described by different domain ontologies introduce the requirement for network-wide semantic integration and distributed service discovery.*

The work presented here describes a distributed and scalable model as a solution to the above problem. We assume that ontologies are composed of concepts and that services are specified using these concepts. Specifically, this paper proposes a model to facilitate progressive ontology matching and concept-based service discovery. At the core of the model is a partial view, called the *concept view*, which nodes maintain over all available concepts and a gossip protocol that periodically disseminates a randomised set 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 semantic knowledge of each node by progressively matching all concepts, and to 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 explain as to how it facilitates 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, and 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 discovery protocols, e.g., UDDI, SOAP, are available to enhance service 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 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 nodes remain connected. It also needs to be taken into account that physical resources e.g., battery life, CPU power, and bandwidth are scarce and require consideration when trying to exchange and match different ontologies.

2.1 Model Description

The model assumes that mobile nodes can be both providers and consumers of services. Once in range to an ad hoc network, each node will execute the gossip protocol, thereby advertising its semantic content. Discovery queries are then initiated on demand by service-based applications that require non-local functionality. Initial identification of other nodes is handled by a specialised bootstrap protocol, which discovers other participants and advertises the new node's presence. The bootstrap procedure can also be used to establish connectivity between two or more nodes that are not yet part of any network. Without loss of generality, each node may maintain multiple ontologies, which are treated as a single ontology by the model.

The model specification uses *concepts* as the core abstraction and *randomisation* as the main process. The specification begins by considering a finite set of concepts that are split uniformly across nodes. The aim is to compare all concepts in a pair-wise fashion by using all available nodes, while at the same time providing a concept replication pattern that scales as both number of nodes and number of concepts increase. The model works on the assumption that both ontologies and semantic queries can be easily decomposed into their constituent concepts. To simplify the description of the model, each node is assumed to know every other node. In reality, the model only requires each node to know a subset of the participating nodes.

Initially, each node randomly selects a fixed number of its concepts and transmits them to a random but fixed number of destination nodes. In subsequent steps, each node produces a

Third IEEE International Conference on
Wireless and Mobile Computing, Networking and Communications (WiMob 2007)
0-7695-2889-9/07 $25.00 © 2007 IEEE

union between the received concepts and those from its own ontology and performs the same random selection and transmission of concepts. To prevent the eventual replication of all concepts into all nodes, the model restricts both the retransmission of received concepts and the distance over which they may propagate. This restriction is necessary to allow the model to scale when the number of concepts increases. The restriction is facilitated with the use of two constraint parameters, a transmission threshold (age) and a propagation threshold (time to live).

The model also requires each participating node to contribute a fraction of its resources for ontology matching and replication. Ontology matching uses the pair-wise concept comparison in tandem with the randomised concept exchange to obtain network-wide semantic agreement in a progressive and distributed fashion. The replication pattern that is derived from concept exchange is also used for concept discovery. In particular, a discovery query follows a random path until it identifies concepts that are similar to the query's constituent concepts. Replication guarantees that similar concepts will be found in a probabilistically bound number of hops.

Such a model offers a scalable mechanism for the discovery of autonomous semantic services and an efficient way to match heterogeneous ontologies. It is scalable because each node maintains only partial semantic metadata and efficient because matching load is progressive and distributed across all nodes. This paper describes the gossip protocol that provides the substrate for discovery. The actual discovery process is elaborated in [20].

3 System Model

This section provides the set of system assumptions and notation used throughout the rest of the paper. The model considers an ad hoc network as composed of a set $\mathcal{N}$ of mobile nodes. A subset $\mathcal{M} = \{n_1, n_2, \ldots, n_{\mathtt{M}}\} \subseteq \mathcal{N}$ of size $\mathtt{M}$ are the participants who maintain ontologies and share services. All nodes are addressed by a unique identifier (id). Nodes that are not participants need only have the basic capability of forwarding packets. In the general case, forwarding can be facilitated using an optimised flooding protocol such as SMURF [24]. In the current implementation, a custom flooding protocol is used that simply rebroadcasts received packets that have not been seen before to all one-hop neighbours. It is assumed that all nodes communicate using a fixed-range, wireless medium e.g., IEEE 802.11b, and that a unicast routing protocol e.g., AODV, OLSR, etc., is available.

Each node $n_i \in \mathcal{M}$ maintains three different views. The views are denoted as follows: $\mathcal{V}_i^{\mathtt{O}}$ for the ontology view, $\mathcal{V}_i^{\mathtt{C}}$ for the concept view, and $\mathcal{V}_i^{\mathtt{N}}$ for the node view.

The ontology view represents a node's ontology as a fixed set of concepts, $\mathcal{V}_i^{\mathtt{O}} = \{c_{i1}, \ldots, c_{i\mathtt{G}}\}$, where $c_{il} \in \mathcal{V}_i^{\mathtt{O}}, 1 \leq l \leq \mathtt{G}$ is a concept in the ontology of source node n_i and $\mathtt{G}$ represents the maximum number of concepts in the ontology view per node. The model assumes that these ontologies are static, so this view allows no additions or deletions of concepts for the duration of the protocol's execution. This assumption is reasonable as ontologies are structured metadata, meaning they are specified during application design and do not change often.

A node's 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 node's ontology view. We represent this view as $\mathcal{V}_i^{\mathtt{C}} \subseteq \{c_{k\mathtt{l}} \mid c_{k\mathtt{l}} \in \mathcal{V}_k^{\mathtt{O}}, k \in \mathcal{M} - \{n_i\}, 1 \leq l \leq \mathtt{G}\}$, where $c_{k\mathtt{l}}$ represents a concept from any ontology view except that of node n_i. The gossip protocol ensures the concept view maintains the following properties. It is:

- *probabilistically bound* – the concept view is not constrained by a fixed size, rather the protocol guarantees that a certain view size is associated with a probability. The intent of the different gossip parameters is to keep the concept view partial, i.e., with a certain probability it should maintain only a subset of the total number of concepts,
- *evolving* – the gossip protocol constantly inserts and removes concepts,
- *a simple random sample* – 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. Furthermore, each view contains a set of concepts that is independent from any other view.

The node view is a set of node identifiers, $\mathcal{V}_i^{\mathtt{N}} \subseteq \{n_k \mid k \in \mathcal{M} - \{n_i\}\}$. Like the concept view it does not allow duplicate node identifiers and can not contain the node's own identifier. It is a membership view that has similar properties to those found in recent gossip protocols [7, 14, 16]. The node view is:

- *fixed-size* – contrary to the concept view, the node view does not automatically adapt to an ever-increasing group size. More sophisticated protocols have been proposed that adapt the size of the membership view as the group size increases [8], but that is outside the scope of this paper. The node view is also intended to be partial, i.e., it does not contain the complete list of all participants,
- *uniform* – the probability that a node identifier exists in a node view is the same for all node identifiers. In other words, if we merge all node views, each node identifier will have the same frequency of appearance,
- *randomised* – the distribution of identifiers in the node views is a random distribution.

The node view is populated during a bootstrap phase and is subsequently maintained by the gossip protocol. The consequence of maintaining the two partial views, $\mathcal{V}^{\mathtt{N}}$ and $\mathcal{V}^{\mathtt{C}}$, is that no node holds complete knowledge of all participating nodes or all ontology concepts. This enables the model to scale in terms of both nodes and concepts and forms the core of the matching and discovery processes.

The gossip protocol is completely characterised by the following parameters:

- $\mathtt{F}_c$ – the concept fanout specifies the number of concepts a source node includes in a gossip message,

Third IEEE International Conference on
Wireless and Mobile Computing, Networking and Communications (WiMob 2007)
0-7695-2889-9/07 $25.00 © 2007 IEEE

- $\mathtt{F}_n$ – the node fanout specifies the number of destination nodes a gossip message is sent to,

- $\mathtt{T}_a$ – the age parameter specifies the number of times a node transmits a received concept before the concept is removed from its concept view. For convenience we define the function $g\_\_(c)$ that takes concept c as input and returns its age,

- $\mathtt{T}_t$ – the time to live (ttl) value is assigned by each source node to any concept that is selected for transmission from its ontology view. It specifies the number of hops that a concept will traverse before being discarded, i.e., the number of retransmissions before the ttl value reaches zero.

3.1 Gossip Protocol

3.1.1 Bootstrapping

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 new node enters an ad hoc network or failed node wishes to rejoin. 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 [5], 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 to allow its node view size to exceed a user-defined threshold.

3.1.2 Gossip Protocol

The gossip protocol described here is executed by each participating node with the aim to facilitate semantic agreement and provide a semantic overlay through concept replication. The specification is given in terms of actions taken during the reception and transmission of a gossip message. A gossip message is used to encapsulate a fixed number of random concepts. The protocol uses a gossip push mechanism to disseminate information and contrary to other gossip protocols in domains such as multicast and failure detection, it randomly replicates a potentially large, but finite set of data items (concepts) across all nodes.

Using the epidemic terminology, we consider each concept as a "virus". Initially, all nodes but one are regarded as susceptible. A gossip transmission from the initial infective node will then "infect" a random but fixed number of other nodes. Subsequent infective nodes maintain each virus (concept) until certain conditions are met in which case the virus is removed. During the period of infection, a node can infect other susceptible nodes. It follows that nodes with a non-empty concept view are infected by multiple viruses and each node is always susceptible if a virus is not in its concept view.

Listing 1 Gossip Transmission

```
1:  //GMsg represents the network packet
2:  Every t ms at node j
3:  Choose {c_k1, ..., c_kFc}, k ∈ M random concepts from V_j^O ⋃ V_j^C
4:  for all c ∈ {c_k1, ..., c_kFc} do
5:    if c ∈ V_j^O then
6:      c.ttl ← Tt
7:    end if
8:    if c ∈ V_j^C ∧ g_(c) = Ta then
9:      V_j^C ← V_j^C − {c}
10:   else if c ∈ V_j^C then
11:     g_(c) ← g_(c) +1
12:   end if
13: end for
14: GMsg.once ← {c_k1, ..., c_kFc}
15: //Reinforcement of membership view
16: Choose a random nid from V_j^N
17: GMsg.nid ← {j, nid}
18: W ← V_j^N − {nid}
19: //Select destination nodes
20: Choose Fn random nodes, {n_1, ..., n_Fn} from W
21: for all nid ∈ {n_1, ..., n_Fn} do
22:   send(nid, GMsg)
23: end for
```

Listing 1 describes the transmission of a gossip message. Periodically, a node selects $\mathtt{F}_c$ concepts uniformly at random from the union of the ontology and concept views (line 3). A concept that is selected from the ontology view is augmented with the ttl property (line 6), signifying the number of hops the concept will be propagated before being dropped by the receiving node. If a concept is selected from the concept view and has already been transmitted $\mathtt{T}_a$ times, it is removed from that view (line 9). Selecting a fixed but random number of concepts from the union of the two views is a simple algorithm that exhibits 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 the ontology view have a higher probability of being selected for transmission.

Each time a node transmits a gossip message, it also piggybacks a single node identifier from its own node view (line 16). The random id together with the identifier of the sender (line 17), serve to reinforce the node view of the receiving node and maintain a uniform distribution even after new admissions and disconnections. The idea is to keep a fluctuating node view across nodes, so that initial cluster effects disappear over time.

On reception of a gossip message, the receiving node executes the algorithm presented in Listing 2. The receiver matches the set of concepts included in the gossip message against its own stored concepts. This is shown in line 5 and is the focus of Section 3.3. Subsequent to matching, lines 6 – 9 show that if a concept is not found in either the concept or the ontology views and the concept's ttl is greater than one, it is stored in the receiver's concept view and the concept's ttl value is decremented by one.

Listing 2 Gossip Reception

```
1:  //Gossip message g represents the gossip packet
2:  On reception of a Gossip message g at node j
3:  //Process received concepts
4:  for all c ∈ Gossip g .once do
5:     Execute ontology matching algorithm between c and
       V_j^O ⋃ V_j^C
6:     if c ∉ V_j^O ∧ c ∉ V_j^C ∧ c.ttl > 1 then
7:        c.ttl ← c.ttl − 1
8:        V_j^C ← V_j^C ⋃ {c}
9:     end if
10: end for
11: //Node view maintenance
12: for all nid ∈ Gossip g .node s do
13:    if nid ≠ j ∧ nid ∉ V_j^N then
14:       if |V_j^N| = Parameter NodeView then
15:          Prune V_j^N by removing a random node id
16:       end if
17:       V_j^N ← V_j^N ⋃ {nid}
18:    end if
19: end for
```

Through the ontology matching algorithm, each message has the potential to create new associations between concepts from different ontologies. Such associations are realised with the use of metadata that are inserted into each concept. This progressive aspect of ontology matching results in the following network behaviour. The longer a node is connected to an ad hoc network, the more likely it is that potential associations between its own and other ontologies will be identified. This prevents nodes that are short-lived or transiently connected from immediately overloading the network with the task of complete ontology matching. A concept view size that is probabilistically bound and a fixed concept fanout also 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 the exchange of concepts and scalable discovery through concept replication.

The last action a receiver undertakes is node view related. A simple algorithm derived from [7] is used, which maintains the view's fixed size while constantly updating it with a small number of new identifiers. It is detailed in lines 12 – 18.

The effect of the randomised concept exchange is twofold: first, the ontology matching component in each node can derive semantic similarity relations between its stored concepts and those received; second, replicated concepts can be used to bound the number of hops required for discovery. The gossip protocol is the basis of the model and provides the foundation on top of which efficient distributed ontology matching and semantic service discovery can take place.

3.2 Illustration

A hypothetical gossip session with gossip parameters $\mathtt{F}_c = 3, \mathtt{F}_n = 1$, $\mathtt{T}_a = 1$, and $\mathtt{T}_t = 2$ is illustrated in Fig. 1. We assume that at round r, node i has an empty concept view and an ontology view that contains three concepts, A, B, and C.

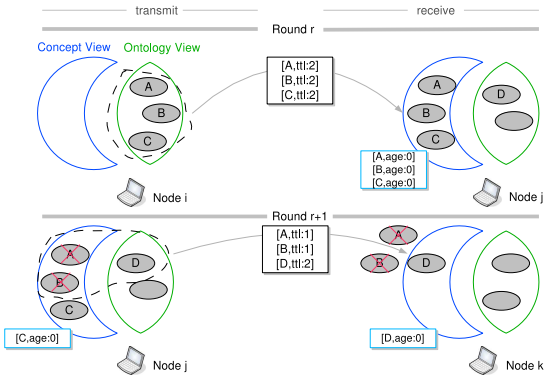

Figure 1. An example operation of the gossip protocol over two rounds. The characteristic parameters are $\mathtt{F}_c = 3$, $\mathtt{F}_n = 1$, $\mathtt{T}_a = 1$ and $\mathtt{T}_t = 2$.

Node j exists in the node view of node i and is selected as the destination for i's gossip transmission. Since all three concepts exist in the ontology view, each one is assigned a new ttl value of $\mathtt{T}_t = 2$. The gossip transmission is received by node j and the three concepts are inserted into the concept view as j contains none of these concepts. With each insertion, the concept's ttl value is decremented by one. An internal table that exists at each participant and maps concepts to their current age is also updated with the name of the concept and a default value of zero.

During round $r+1$ we observe node j. It is assumed that node k exists in j's node view and that k is selected as the gossip destination. Furthermore, the concept selection process yields concepts A, B, and D as the gossip message content. Since the age parameter is set to $\mathtt{T}_a = 1$, once a concept gets selected from the concept view it is also removed. Hence, after transmission, only concept C remains in j's concept view, with A and B being removed. Note however that any matching associations that were established between A, B, or C and j's ontology view concepts continue to persist. They will only be removed once i disconnects. At the destination side, node k examines the ttl values of the received concepts and *after* matching A, B, and D with its own concepts, inserts D in its concept view, since it is the only one with a ttl value greater than one. Similar to the previous reception by node j, an entry in the age table is created for the newly added concept.

3.3 Ontology Matching

Many ontology matching techniques have been proposed in the literature [6, 3]. However, specific properties of the MANET environment and of 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

Third IEEE International Conference on
Wireless and Mobile Computing, Networking and Communications (WiMob 2007)
0-7695-2889-9/07 $25.00 © 2007 IEEE

a progressive matching approach,

3. *randomisation* – detaches concepts from their semantic context. The gossip protocol selects concepts for transmission independent of their adjacent concepts. As such, the context of a concept, which is defined by its adjacent concepts in the ontology graph is often not transmitted. This makes challenging the usage of matching techniques like the deep and intensive model in [3], where the context of a concept is used to produce 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 to *intermediate matching* of the H-Match algorithm. A match between two concepts is recorded when their respective names correspond and all of their properties match in both 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 demands certain requirements from the concept description. Ontology languages such as RDFS and OWL were not designed for this task so our model requires an augmented syntax for concepts that are distributed across a network.

We call this syntax the *network representation* of a concept, to distinguish it from its normal representation in a local ontology. In short, the network representation aids pair-wise concept matching and discovery of services by including references to concept's properties and source node identifiers. An instance of a concept's network representation is 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>
```

There is a dependency between a concept's network representation and the accuracy of the matching algorithm. A network representation that contains only concept names will make it difficult for any matching algorithms to produce an accurate match. In the current prototype the network representation of each concept includes references not only to the set of concept properties but also to the properties inherited from its super concepts. This increases the per concept information, which in turn enables more accurate concept matching.

Semantic similarity is established with the use of a transitive and symmetric relation between two concepts. This occurs only after the concept matching process has compared the two concepts and has identified that they are semantically equivalent. If c_{ix}, c_{jy}, and c_{kz} represent concepts in $\mathcal{V}_i^0, \mathcal{V}_j^0, \mathcal{V}_k^0$; $\mathbf{M}$ represents the matching relation; and a match exists between c_{ix}, c_{jy} and also between c_{jy}, c_{kz}, the following matches are also established:

1. $c_{ix}\mathbf{M}c_{jy} \Rightarrow c_{jy}\mathbf{M}c_{ix}$ (symmetry)
2. $c_{jy}\mathbf{M}c_{kz} \Rightarrow c_{kz}\mathbf{M}c_{jy}$ (symmetry)
3. $c_{ix}\mathbf{M}c_{jy} \wedge c_{jy}\mathbf{M}c_{kz} \Rightarrow c_{ix}\mathbf{M}c_{kz}$ (transitivity)

The exact semantics of the matching relation can vary and ultimately depend on the strength of the matching mechanism. For this paper, the scope of matching is reduced to an equivalence relation that is similar to the `owl:equivalentClass` property in OWL. Other relation types are also possible, such as the `owl:kindOf` or `owl:partOf` OWL predicates.

Fig. 2 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 both the names and properties of concepts results in a bidirectional matching relationship, in this case between the two 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.

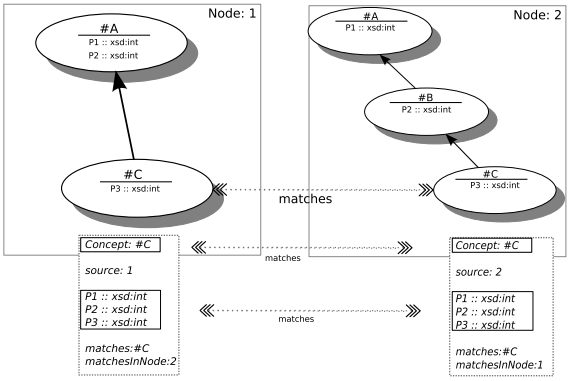

Figure 2. Creating matching associations between concepts.

Establishing these kind of transitive and symmetric relations in a decentralised way can easily lead to inconsistencies. Matching ambiguities can arise especially if a more flexible matching algorithm is used, e.g., one that uses a thesaurus or Wordnet [18], instead of strict string matching. For example, consider the distinct concepts $c_{ix}, c_{iy} \in \mathcal{V}_i^0$ and concept $c_{jz} \in \mathcal{V}_j^0$. It is possible for a node to match c_{ix} with c_{jz}, while another node matches concept c_{jz} with c_{iy}. Eventually, the transitive relation $c_{ix}\mathbf{M}c_{jz}\mathbf{M}c_{iy}$ will be established in nodes n_i and n_j. However, the two distinct concepts in node n_i are now matched through concept c_{jz}. This problem can appear when matching individual concepts rather than complete ontologies.

Aside from practical solutions to this problem, e.g., use of a matching relation that is not transitive, such inconsistencies do not affect the correctness of the model. The discovery process

Third IEEE International Conference on
Wireless and Mobile Computing, Networking and Communications (WiMob 2007)
0-7695-2889-9/07 $25.00 © 2007 IEEE

only requires the identity of a node from a matching relation and not the actual name of the matching concept. This simplifies the discovery algorithm and overcomes the above inconsistencies.

4 Implementation and Evaluation

The gossip protocol was implemented using the ns-2 simulator. We used RDF(S) for describing the individual ontologies as it is more lightweight than OWL-S, albeit with weaker inference semantics. The simulation uses an area of $1500m \times 500m$ with the cardinalities of node sets $\mathcal{N}$ and $\mathcal{M}$ being 70 and 42 respectively. For the mobility model we selected a steady-state random waypoint model [22] for more accurate simulation results. Each node ontology is composed of 10 concepts, bringing the total number of concepts to 420.

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 algorithm to find a node that contains the queried concept. Specifically, each node searches both its concept and ontology views for the concepts that compose the discovery query. If the concepts are not found, another target node is selected randomly from the node view and the discovery query is retransmitted to that target node. The second evaluation criterion, ontology matching latency, is specified relative to the number of rounds required for complete semantic agreement between all available ontologies.

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 (T_a) and ttl (T_t). Since the goal of this 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 [7] 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, which is encapsulated in a UDP packet.

Figure 3 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 and hence discoverability, but requires increased storage and processing resources.

Figure 4 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. The actual matching approach used for this experiment establishes associations only between the received concepts and concepts in the ontology view of each node. Although this increases the matching latency as compared to an approach that matches received concepts with concepts in both the ontology and concept views, it reduces processing overhead and duplication of effort. As

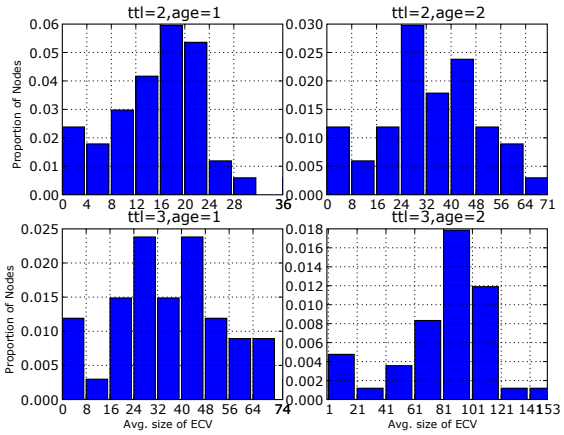

Figure 3. The distribution of concept view sizes between nodes.

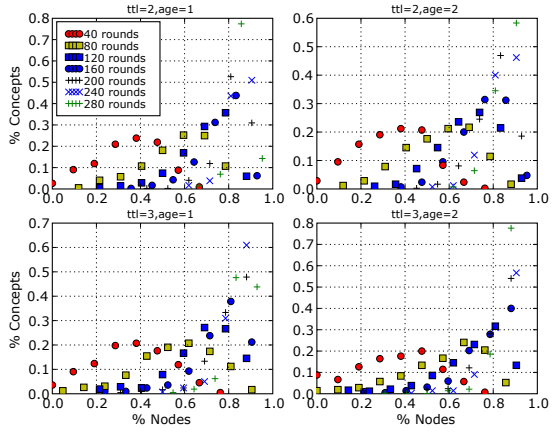

Figure 4. Infection spread as concepts infect nodes.

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.

Figure 5 displays the matching latency as each received concept is matched against both the concept and ontology views. The figure illustrates that pair-wise matching can be achieved very quickly with the age and ttl parameters having minor influence.

Finally, Fig. 6 plots the concept discovery performance as a function of the number of hops required before a single 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 concept

Third IEEE International Conference on
Wireless and Mobile Computing, Networking and Communications (WiMob 2007)
0-7695-2889-9/07 $25.00 © 2007 IEEE

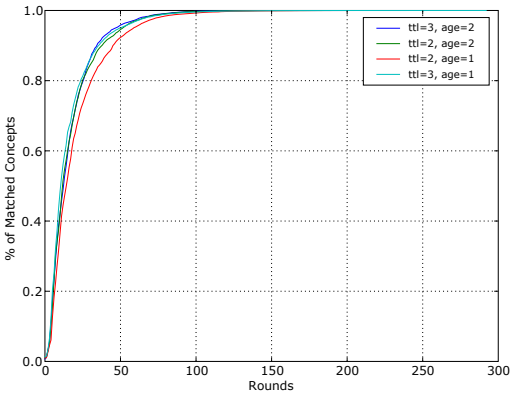

Figure 5. Percentage of pair-wise concept matching versus rounds.

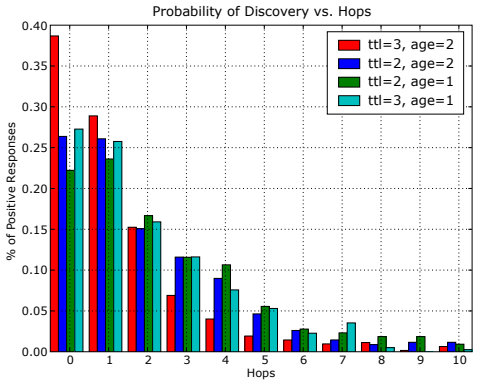

Figure 6. Distribution of concept discovery vs. number of hops required.

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 with maintaining the overlay network in each participant's node view, most of the optimisations applied in [15] 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 as it would reduce the need for constant route discoveries.

The protocol also requires an explicit leave operation, whereby a node wishing to depart multicasts a leave request to several participants chosen from its 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 present 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., [14], [27], [13], [12]. However, this assumed strict a priori agreement 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 exchange 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 [26], [23], and [25] have provided a methodology for the semantic matchmaking of services in the context of the web. In particular, [23] 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 [21], assume that not only is data distributed, but that metadata descriptions are also decentralised and not uniform. As a broader set of assumptions can be made about resource availability and peer failure rates in P2P networks, more sophisticated schema mapping techniques are feasible. A practical model for ontology mapping is presented in [3].

Finally, gossip protocols have been used in many network topologies to achieve a tradeoff between efficiency and reliability. Specifically, in [15], a gossip protocol is presented that provides probabilistic reliability with good scalability properties in a MANET environment. In [7], a gossip protocol is developed in the context of a P2P publish-subscribe system that provides a partial but fixed membership view of the participating nodes. In our model the node view is modelled after the partial membership model 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 pair-wise compared. Multicast gossip protocols usually involve the transmission of potentially infinite number of packets to all participating nodes.

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 to facilitate the emergence of a shared ontology, while offering efficient 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.

A stochastic analysis of the gossip protocol presented in this paper has been concluded and can be found in [19]. As part of the analysis, a Markov-based model was constructed in order to verify the scalability of the protocol. The Markov model aims to predict the memory consumption, processing overhead and the probabilistic guarantees for service discovery. The analytical results verify the results of the simulation, further validating our claims.

References

[1] K. Aberer, P. Cudré-Mauroux, A. M. Ouksel, T. Catarci, M.-S. Hacid, A. Illarramendi, V. Kashyap, M. Mecella, E. Mena, E. J. Neuhold, O. De Troyer, T. Risse, M. Scannapieco, F. Saltor, L. De Santis, S. Spaccapietra, S. Staab, and R. Studer. Emergent Semantics Principles and Issues. In *DASFAA 2004*, pages 25–38, 2004.

[2] K. Arnold, R. Scheifler, J. Waldo, B. O'Sullivan, and A. Wollrath. *Jini Specification*. Addison-Wesley Longman Publishing Co., Inc., 1999.

[3] S. Castano, A. Ferrara, and S. Montanelli. H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems. In *SWDB*, pages 231–250, 2003.

[4] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha. 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, September 2002.

[5] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry. Epidemic Algorithms for Replicated Database Maintenance. In *Proceedings of the 6th annual ACM Symposium on Principles of Distributed Computing (PODC'87)*, pages 1–12, New York, NY, USA, 1987. ACM Press.

[6] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of the 11th international conference on World Wide Web (WWW '02)*, pages 662–673, New York, NY, USA, 2002. ACM Press.

[7] P. T. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A. M. Kermarrec. Lightweight Probabilistic Broadcast. *ACM Transactions on Computer Systems*, 21(4):341–374, 2003.

[8] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulie. SCAMP: Peer-to-Peer Lightweight Membership Service for Large-Scale Group Communication. In *Third International Workshop on Networked Group Communications (NGC 2001)*, London, UK, November 2001.

[9] J. Heflin. OWL Web Ontology Language Use Cases and Requirements. W3C Recommendation, 2004.

[10] S. Helal, N. Desai, V. Verma, and C. Lee. 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, March 2002.

[11] Internet Assigned Numbers Authority (IANA). *Service Location Protocol Extensions, Version 2 (SLPv2)*, May 2003.

[12] U. C. Kozat and L. Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *IEEE INFOCOM*, volume 22, pages 1965–1975, March 2003.

[13] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 120–130. ACM Press, 2000.

[14] J. Luo, P. T. Eugster, and J. P. Hubaux. Route Driven Gossip: Probabilistic Reliable Multicast in Ad Hoc Networks. In *Proceedings of the 22nd IEEE INFOCOM*, pages 2229–2239, 2003.

[15] J. Luo, P. T. Eugster, and J. P. Hubaux. Probabilistic Reliable Multicast in Ad Hoc Networks. *Ad Hoc Networks*, 2(4):369–386, 2004.

[16] Luo, Jun and Eugster, Patrick Th. and Hubaux, Jean-Pierre. PILOT: ProbabilistIc Lightweight grOup communication sysTem for Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 3(2):164–179, 2004.

[17] Microsoft Corporation. *Universal Plug and Play: Background*, 1999.

[18] G. A. Miller. Wordnet: a Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[19] A. Nedos. *Discovery of Autonomous Semantic Services in Mobile Ad Hoc Networks*. PhD thesis, Distributed Systems Group, Department of Computer Science, Trinity College Dublin, 2007.

[20] A. Nedos, K. Singh, and S. Clarke. Mobile Ad Hoc Services: Semantic Service Discovery in Mobile Ad Hoc Networks. In A. Dan and W. Lamersdorf, editors, *Proceedings of the 4th International Conference on Service-Oriented Computing (ICSOC)*, volume 4294 of *Lecture Notes in Computer Science*, pages 90–103. Springer, Dec. 2006.

[21] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: A P2P Networking Infrastructure Based on RDF. In *Proceedings of the 11th International Conference on World Wide Web (WWW'02)*, pages 604–615, New York, NY, USA, 2002. ACM Press.

[22] S. PalChaudhuri, J.-Y. L. Boudec, and M. Vojnovic. Perfect Simulations for Random Trip Mobility Models. In *Proceedings of the 38th annual Symposium on Simulation (ANSS'05)*, pages 72–79, Washington, DC, USA, 2005. IEEE Computer Society.

[23] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara. Semantic Matching of Web Services Capabilities. In *Proceedings of the 1st International Semantic Web Conference ISWC'02*, pages 333–347, London, UK, 2002. Springer Verlag.

[24] C. E. Perkins, T. Clausen, and P. Jacquet. *Multicast With Minimal Congestion Using Connected Dominating Sets*. IETF, 2005.

[25] K. Sycara, S. Widoff, M. Klusch, and J. Lu. LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5(2):173–203, June 2002.

[26] D. Trastour, C. Bartolini, and J. Gonzalez-Castillo. A Semantic Web Approach to Service Description for Matchmaking of Services. In I. F. Cruz, S. Decker, J. Euzenat, and D. L. McGuinness, editors, *SWWS*, pages 447–461, 2001.

[27] Y. Xue, B. Li, and K. Nahrstedt. A scalable location management scheme in mobile ad-hoc networks. In *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks*, pages 102–112. IEEE Computer Society, 2001.