

Probabilistic Discovery of Semantically Diverse Content in MANETs

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

Abstract—Mobile ad hoc networks rely on the opportunistic interaction of autonomous nodes to form networks without the use of infrastructure. Given the radically decentralized nature of such networks, their potential for autonomous communication is significantly improved when the need for a priori consensus among the nodes is kept to a minimum. This paper addresses an issue within the domain of semantic content discovery, namely, its current reliance on the preexisting agreement between the schema of content providers and consumers. We present OntoMobil, a semantic discovery model for ad hoc networks that removes the assumption of a globally known schema and allows nodes to publish information autonomously. The model relies on the randomized dissemination and replication of metadata through a gossip protocol. Given schemas with partial similarities, the randomized metadata dissemination mechanism facilitates eventual semantic agreement and provides a substrate for the scalable discovery of content. A discovery protocol can then utilize the replicated metadata to identify content within a predictable number of hops using semantic queries. A stochastic analysis of the gossip protocol presents the different trade-offs between discoverability and replication. We evaluate the proposed model by comparing OntoMobil against a broadcast-based protocol and demonstrate that semantic discovery with proactive replication provides good scalability properties, resulting in a high discovery ratio with less overhead than a reactive nonreplicated discovery approach.

Index Terms—Mobile ad hoc networks, distributed discovery, probabilistic algorithms, gossip protocols, semantic services.

1 INTRODUCTION

MOBILE Ad Hoc Networks (MANETs) are formed when autonomous mobile devices with short-range wireless communication capabilities cooperate to provide spontaneous connectivity. Although a challenging environment, MANETs find wide applicability in scenarios where opportunistic networking and collaborative activity is required, e.g., pervasive computing and mobile games. In such a dynamic environment, the location of required content (i.e., services or data) cannot be hard-wired because nodes can acquire transient addresses and networks may partition at any time. The result is that discovery becomes an important process preceding any collaborative effort. In general, discovery attempts to match required and available functionality by using an appropriate representation language and suitable network support.

In MANETs, representation and discovery of content has several unique requirements: the opportunistic nature of the network necessitates reduced human intervention requiring automated discovery; node autonomy implies that content representation is difficult to standardize or to guarantee its agreement between providers and consumers; the open and dynamic nature of the network requires discovery mechanisms that scale; and resource-constrained devices dictate the efficient distribution of discovery load. We elaborate on each of these requirements below.

- The authors are with the Distributed Systems Group, Department of Computer Science, School of Computer Science and Statistics, Trinity College Dublin, Dublin 2, Ireland.
E-mail: {Andronikos.Nedos, Raymond.Cunningham, Siobhan.Clarke}@cs.tcd.ie, zapfmann@gmail.com.

Manuscript received 8 Nov. 2007; revised 9 May 2008; accepted 18 Aug. 2008; published online 5 Sept. 2008.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2007-11-0336. Digital Object Identifier no. 10.1109/TMC.2008.133.

When minimum user intervention is required in mobile networks with no stable and permanent set of nodes, manually identifying relevant functionality requires user involvement and constitutes additional discovery latency. Representing services or data with capabilities rather than names offers greater flexibility in discovery queries. A more expressive content description can encourage automated discovery based on queries about explicit functionality [1], rather than discovery based on the implicit association between syntactic interfaces and content. Semantic services are a good example of how ontologies can facilitate a semantic interface for discovery and also represent the capabilities of services.

A second requirement for content discovery is the necessary agreement on the information representation between providers and clients. Standardization of information schemas is one way to address this issue, though the administrative overhead is not negligible and requires centralized infrastructure and planning. The problem of having a standardized representation is exacerbated in MANETs when one considers the self-contained and unpredictable environment. As MANETs can form in places where no Internet connectivity is guaranteed, it follows that nodes cannot avail of common representations on the Internet and can only use the knowledge of connected nodes. When ontologies provide the representation language, it is not reasonable to assume that information in mobile nodes will be described using commonly agreed ontologies. Rather, the use of multiple and independently developed domain ontologies is more likely. A comprehensive specification of this problem is given in [2], where the authors use the term *emergent semantics* to describe semantic consensus based not on standardization but on emergent behavior.

In the distributed context envisioned by the emergent semantics model, maintaining semantic interoperability remains a strong requirement. Even when data and services are described by multiple and independently developed ontologies, service discovery, content retrieval, and semantic inference should be guaranteed as if operating within a single ontology. To this end, appropriate mechanisms are required to map or translate metadata [3], [4].

Projects like H-MATCH [5], LARKS [6], and GLUE [4] accept semantic heterogeneity and have devised techniques and algorithms for ontology matching, mapping, and evolution.

The additional challenges that MANETs pose to the discovery of semantically diverse content relate to scalability and efficiency. Open networks with nodes that can be both content providers and consumers require discovery protocols that scale. Furthermore, mobility and limited device resources dictate that selected techniques must be adaptable, configurable, and able to spread load efficiently. In general, application interoperability in MANETs faces issues that require careful consideration of established assumptions such as reliance on standardized syntactic interfaces or availability of common ontologies.

Current discovery architectures for MANETs address the above requirements to varying degrees. Traditional service discovery architectures like Sun's Jini [7] and IETF's Service Location Protocol (SLP) [8] were not designed for MANET environments, so they scale poorly and require global knowledge of service templates. Other approaches have concentrated on distributed discovery protocols [9], [10], [11] but limit node autonomy by assuming a simple representation language globally known by all mobile peers.

The contribution of this paper lies in the design, analysis, and evaluation of a distributed discovery model called *OntoMobil*. *OntoMobil* caters for MANETs and semantic decentralization. Semantic decentralization is the idea that autonomous nodes can express data or services using different ontologies that are not agreed upon a priori. The model relies on the decomposition of ontologies into concepts and the replication of these concepts through a gossip protocol. This randomized concept dissemination works in tandem with a lightweight semantic matching mechanism that is executed in each node to facilitate the eventual semantic agreement between diverse ontologies and provide a substrate for the scalable discovery of content. The actual discovery employs a random walk protocol, whereby semantic queries are first routed to a number of random nodes and are subsequently evaluated by a semantic reasoner at any provider node with a compatible ontology. A stochastic analysis is used to predict performance and illustrate the trade-offs between high discoverability and the overhead incurred by replication. A comparison between the analytical model and an implementation of *OntoMobil* in ns2 confirms the derived analytical bounds.

The paper has the following layout. Section 2 contains the state of the art in decentralized and mobile discovery systems. Section 3 describes the model and protocols that comprise *OntoMobil*. Section 4 provides the stochastic

analysis of the gossip protocol and probabilistic bounds of the random walk discovery protocol, while the evaluation is presented in Section 5, and the conclusion is given in Section 6.

2 RELATED WORK

The importance of content discovery in the design of flexible and adaptive applications can be seen in the proliferation of discovery architectures. This section reviews existing work with similar goals and features to *OntoMobil* in decentralized semantic topologies and ad hoc discovery protocols.

2.1 Semantic P2P Topologies

The assumption of a centralized architecture and an intuitive scheme to name available resources can simplify the discovery process. The activity of looking through yellow pages or initiating a google query are typical examples of this. When centralized architectures cannot be supported, specialized architectures and protocols are required. A characteristic example is peer-to-peer (P2P) networks. Initially, discovery in these systems was confined to naming conventions with limited expressiveness, putting the emphasis on topology properties such as scalability and bounded routing (e.g., Chord and Pastry). Currently, a new generation of P2P networks uses semantic representations coupled with decentralized topologies.

One of the first complete systems to explore the idea of semantic or schema-based topologies is the *EDUTELLA* project [12]. The goal of *EDUTELLA* is the distributed discovery of semantic information without the use of common ontologies. In terms of topology, *EDUTELLA* is based on superpeers, though the clustering algorithm it uses has several flavors.

Haase et al. [13] describe another semantic P2P topology focused on efficiency. The authors start from a completely random network topology and eventually derive a topology that closely matches the similarity in the semantic knowledge of peers. The authors evaluate the hypothesis that certain semantic topologies will perform more efficiently than random topologies. A number of assumptions underlie this hypothesis. Chief among them are the assumptions of semantic similarity, stable connectivity, and global semantic knowledge.

INGA [14] exploits the query history in order to route semantic queries in P2P networks. It accepts that a common schema between peers is a simplistic assumption and assumes that peers describe content with heterogeneous metadata. Since *INGA* does not define an exact topology, it selects a number of peers that are likely to contain relevant results through observation and recording of meta-information from user queries. This progressive acquisition of knowledge is a common characteristic between *INGA* and *OntoMobil*, though the two systems use very different methods.

OntoMobil differs from existing semantic topologies in its assumptions, its topology, and the guarantees it provides. First, *OntoMobil* is targeted toward MANETs, which are inherently dynamic, composed of autonomous nodes, and characterized by transient connectivity. These

conditions prevent solutions that rely on stable connections and global semantic knowledge. Second, OntoMobil builds an unstructured semantic overlay by maintaining a replicated set of concepts. Although the OntoMobil topology is not intended to be a general-purpose P2P topology, it can still be classified according to the SIL model [15] as a graph with *forwarding search* and *forwarding index* links. What further differentiates the OntoMobil topology in relation to other general-purpose content addressable topologies, e.g., PlanetP [16], is the use of multiple weakly consistent replicated indexes in the form of concepts instead of a single index. Finally, the use of epidemic updates to maintain concepts weakly consistent has the main benefits of performance, simplicity, and an algorithm amenable to an analytical model and predictable guarantees.

2.2 Service Discovery in MANETs

Despite similarities between P2P networks and MANETs, there are a number of important differences. In an environment where mobile nodes develop content without a common schema, ontology matching and discovery must be robust to node mobility, consider the processing capacity of nodes, and adapt to variation in the size of the network. We illustrate these design trade-offs by comparing OntoMobil to the following discovery systems in MANETs.

Kozat and Tassiulas [9] present a distributed discovery mechanism for MANETs. The mechanism is broker-based and operates very close to the routing layer. There is no suggestion as to what type of services can be supported in such a system as the focus is on the specification of the discovery protocol and its performance evaluation. The broker-based topology designed by Kozat and Tassiulas and the semantic overlay created by OntoMobil satisfy different requirements, making a direct comparison difficult. If the broker-based topology were to be used to fulfill the requirements of OntoMobil, the broker nodes would have to bear the sole responsibility for concept matching. This would increase their processing overhead and would also require a specialized hand-off procedure to transfer the matching relations between departing and new broker nodes. Because of mobility, the hand-off procedure and the constant swapping of nodes between brokers and non-brokers would increase the traffic overhead.

Sailhan and Issarny [10] describe a scalable service architecture for mobile ad hoc and hybrid networks. The architecture is composed of a service representation and a suite of protocols for advertisement and service discovery. The requirement of reduced energy consumption have led to a design that incorporates a set of connected broker nodes forming an overlay network. The core differences between the OntoMobil overlay and the discovery architecture in [10] reside in the representation of content, the fabric of the overlay network, and the use of broker nodes. Because OntoMobil uses ontologies to describe content, the overlay is constructed from metadata, rather than data or services. Furthermore, in OntoMobil every node is considered to be part of the overlay, which simplifies the design by not requiring a special protocol for node to broker communication.

The Group-based Service Discovery (GSD) protocol [17] uses a caching mechanism to improve access times and

reduce the overhead associated with semantic service discovery in ad hoc networks. The caching mechanism works by employing an ontology-based classification for all services. The use of cached service concepts to aid discovery in GSD resembles that of OntoMobil. Among the differences are the common ontology assumed by GSD and the fact that OntoMobil does not use flooding but relies on a unicast gossip protocol for concept dissemination. This has the potential to improve OntoMobil's scalability properties and to reduce overhead for nodes that do not participate in the exchange of services. GSD also sketches a service matching approach, which functions by identifying service groups, input-output parameters, and service capabilities as potential matchmaking properties. This can prove suboptimal when the selective forwarding strategy is different to the matchmaking one. For example, as outlined in the design of the forwarding strategy, service groups are sufficient to route a service to a provider. This does not guarantee a matched service, however, as the query signature might still be different to the provided one. OntoMobil addresses this concern by routing requests according to their full semantic signature.

3 ONTOMOBIL: MODEL AND PROTOCOLS

This section presents the OntoMobil model and the detailed specification of the gossip and discovery protocols.

3.1 Model Description

The model specification uses *concepts* as the core abstraction and *randomization* 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 pairwise fashion by using all available nodes, while at the same time providing a replication pattern to facilitate rapid discovery. The model uses the intuition that both ontologies and semantic queries are easily decomposed into their constituent concepts. To simplify this abstract model description, each node is assumed to know every other node, and protocol events in all nodes are assumed to be synchronized. In reality, the model only requires each node to know a subset of the participating nodes, while the synchrony assumption is relaxed.

Initially, each node randomly selects a fixed subset of its concepts and transmits these concepts to a set of random nodes. In subsequent steps, each node mixes any received concepts with those from its own ontology and performs the same random selection and transmission. To prevent the eventual replication of all concepts into all nodes, the model restricts both the retransmission of received concepts and their propagation. This restriction is necessary to allow the model to scale as the number of concepts increase. The restriction is facilitated by the use of two constraint parameters: a transmission threshold (age) and a propagation threshold (time to live (ttl)).

The model requires each participating node to contribute a fraction of its resources for ontology matching and replication. Ontology matching combines the pairwise concept comparison with the randomized concept dissemination to obtain network-wide semantic agreement in a progressive and distributed fashion. By utilizing the

infrastructure that emerges from replication and semantic matching, a discovery query can now follow a random path through the one-hop neighborhood of each node until it identifies concepts that are similar to the query's constituent concepts. Replication guarantees that similar concepts will be found in a number of hops that is probabilistically bound.

Such a model offers a scalable mechanism for the discovery of diverse semantic content and an efficient way to match heterogeneous ontologies. It is scalable because each node maintains only partial metadata and efficient because matching load is progressive and distributed across all nodes.

3.2 System Assumptions

We consider an ad hoc network composed of $\mathcal{N} = \{n_1, n_2, \dots, n_N\}$ mobile nodes. All nodes are considered to be active participants that maintain ontologies and share content. The model can be extended to include nodes that are not participants, though they are required to have the basic capability to forward packets. It is assumed that all nodes communicate using a fixed-range wireless medium (e.g., IEEE 802.11) and that a unicast routing protocol (e.g., AODV and OLSR) is available.

Each node $n_i \in \mathcal{N}$ maintains three different views. The ontology view, the concept view, and the node view. The ontology view represents a node's ontology as a fixed set of concepts, $\mathcal{V}_i^O = \{c_{i1}, \dots, c_{iG}\}$, where $c_{il} \in \mathcal{V}_i^O$, $1 \leq l \leq G$ is a concept in the ontology of node n_i , and G represents the maximum number of concepts 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 execution. We believe that this is a reasonable assumption as ontologies are structured metadata, meaning that they are specified during application design and do not change often.

The concept view includes the set of concepts that are received from other nodes. It does not allow duplicate concepts or concepts that exist already in the node's ontology view. This view is represented as $\mathcal{V}_i^C \subseteq \{c_{kl} | c_{kl} \in \mathcal{V}_k^O, k \in \mathcal{N} - \{n_i\}, 1 \leq l \leq G\}$, where c_{kl} represents a concept from any ontology view except the ontology view of node n_i . The gossip protocol provides the concept view with the following properties:

- *Probabilistically bound.* The concept view is not constrained by a fixed size; rather, the protocol guarantees a bound on its size with a certain 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.* Since each view does not contain a set of concepts that concretely describe a knowledge domain, it rather contains randomized concepts that can belong to any of the available ontologies.

The node view is a set of node identifiers, $\mathcal{V}_i^N \subseteq \{n_k | k \in \mathcal{N} - \{n_i\}\}$. Like the concept view, it does not allow

duplicate node identifiers and cannot contain the node's own identifier. It is a membership view that has similar properties to those found in recent gossip protocols [18], [19]. The node view has the following properties:

- *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 [20]. The node view is also intended to be partial without containing the complete list of all participants.
- *Uniform.* The probability that a node identifier exists in a specific node view is the same for all node identifiers.
- *Randomized.* The distribution of identifiers in the node views is that of a random distribution.

The node view is populated during a bootstrap phase and is subsequently maintained by the gossip protocol.

3.2.1 Gossip-Based Parameters

The gossip protocol is completely characterized by the following parameters:

- F_c —the concept fanout specifies the number of concepts a sender includes in a gossip message.
- F_n —the node fanout specifies the number of destination nodes a gossip message is sent to.
- T_a —age specifies the number of times a node transmits a received concept before the concept is removed from the concept view. For convenience, we define function $age(c)$, which takes concept c as input and returns its age.
- T_t —the ttl value is assigned by each sender to any concept that is selected for transmission from its ontology view. It specifies the number of hops in the semantic overlay that a concept will traverse before being discarded, i.e., when the ttl value reaches zero.

3.3 Constructing the Semantic Overlay

3.3.1 Join

Before a node begins execution of the gossip protocol, it must first populate its node view with a partial list of other participants. A simple bootstrap protocol is used in order to reach a partial and uniform membership view. The OntoMobil join protocol is based on a simple *expanding ring search* and is used either when a node enters an ad hoc network or after a node failure.

3.3.2 Leave

The current gossip specification requires an explicit disconnection procedure and does not admit unexpected failures. Before a node leaves the network, it declares its intent as part of the periodic gossip transmission. Nodes that receive such a gossip transmission 1) remove the node identifier if it exists in their node view, 2) remove any concepts from their concept view that originate from the ontology view of the departing node, and 3) remove all references from concepts maintained in their ontology or concept views that point to concepts in the ontology view of the departing node.

In the next gossip round, the nodes that initially received the leave notification will further propagate it, resulting in more nodes executing the same removal sequence as above. Eventually, by gossiping the intention to disconnect, all participants will be reached, and all network-wide references to the departing node will be removed.

3.3.3 Gossip Protocol

The OntoMobil gossip protocol is influenced by [19], retaining similar semantics for the ttl and age parameters and incorporating the use of a partial membership view mechanism found in similar protocols. It has been modified to a distinct protocol, described in more detail in [21], to fit the discovery model of OntoMobil by altering the transmit and receive operations, adding the concept fanout parameter, and optimizing the OntoMobil gossip for a proactive routing protocol. In particular, what distinguishes the two main gossip operations of transmission and reception in [19] is their operation on a finite set of concepts rather than on application-generated messages; the transmission of a randomized selection from the union of the concept and ontology views rather than the transmission of buffered messages until such buffer is empty; and ultimately, the provision of randomized replication rather than randomized transmission.

Listing 1. Gossip Transmission

```

1: //GossipMessage represents the network packet
2: Every  $t$  time units at node  $j$ 
3: Choose  $F_c$  random concepts  $\{c_{k1}, \dots, c_{kF_c}\}$  from  $\mathcal{V}_j^O \cup \mathcal{V}_j^C$ 
   where  $k \in \mathcal{N}$ 
4: for all  $c \in \{c_{k1}, \dots, c_{kF_c}\}$  do
5:   if  $c \in \mathcal{V}_j^O$  then
6:      $c.ttl \leftarrow T_t$ 
7:   end if
8:   if  $c \in \mathcal{V}_j^C \wedge age(c) = T_a$  then
9:      $\mathcal{V}_j^C \leftarrow \mathcal{V}_j^C - \{c\}$ 
10:  else if  $c \in \mathcal{V}_j^C$  then
11:     $age(c) \leftarrow age(c) + 1$ 
12:  end if
13: end for
14:  $GossipMessage.concepts \leftarrow \{c_{k1}, \dots, c_{kF_c}\}$ 
15: //Piggyback removal notifications
16:  $GossipMessage.removals \leftarrow Buffer_{Removals}$ 
17: //Select destination nodes
18: Choose  $F_n$  random nodes,  $\{n_1, \dots, n_{F_n}\}$  from  $\mathcal{V}_j^N$ 
19:  $GossipMessage.src \leftarrow \{j\}$ 
20: for all  $nid \in \{n_1, \dots, n_{F_n}\}$  do
21:    $send(nid, GossipMessage)$ 
22: end for

```

Listing 1 describes the transmission of a gossip message. Periodically, a node selects 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 T_a times, it is removed from the concept view (line 9). Selecting a fixed number of random concepts from the union of the two

views is a simple algorithm that exhibits a desirable adaptive behavior. 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.

If the sender has received any leave notifications, they are also appended to the gossip message (line 16). The last action for the sender is to select F_n distinct nodes uniformly at random from its node view (line 18) and to unicast the gossip message to the target nodes (line 21).

On reception of a gossip message, the receiving node executes the algorithm presented in Listing 2. If the optional field $GossipMessage.removals$ exists, then the actions outlined in Section 3.3.2 are executed. The identifiers are then stored in the auxiliary buffer $Buffer_{Removals}$, so they can be further propagated in the next gossip transmission (line 6).

Next, the receiver matches the set of concepts included in the gossip message against its own stored concepts (line 10). We use the H-MATCH algorithm [5] to identify concept equivalence relationships. When two concepts are found to be equivalent, a reference to the matching concept and its corresponding source node address are appended in both concepts. The current implementation uses RDFS to model ontologies, so two RDFS predicates are appended in both concepts: `om:matchesConcept` holds a reference to the URI of the matching concept, and `om:matchesConceptInNode` references the corresponding node address.

Subsequent to matching, lines 11-14 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: //GossipMessage represents the gossip packet
2: On reception of a GossipMessage at node  $j$ 
3: //Prune entries from removed nodes
4: for all  $nid \in GossipMessage.removals$  do
5:   Execute algorithm in Section 3.3.2
6:    $Buffer_{Removals} \leftarrow Buffer_{Removals} \cup \{nid\}$ 
7: end for
8: //Process received concepts
9: for all  $c \in GossipMessage.concepts$  do
10:  Execute ontology matching algorithm between  $c$  and
    $\mathcal{V}_j^O \cup \mathcal{V}_j^C$ 
11:  if  $c \notin \mathcal{V}_j^O \wedge c \notin \mathcal{V}_j^C \wedge c.ttl > 1$  then
12:     $c.ttl \leftarrow c.ttl - 1$ 
13:     $\mathcal{V}_j^C \leftarrow \mathcal{V}_j^C \cup \{c\}$ 
14:  end if
15: end for
16: //Node view maintenance
17: if  $GossipMessage.src \notin \mathcal{V}_j^N$  then
18:  if  $|\mathcal{V}_j^N| = Parameter_{NodeView}$  then
19:    Prune  $\mathcal{V}_j^N$  by removing a random node id
20:  end if
21:   $\mathcal{V}_j^N \leftarrow \mathcal{V}_j^N \cup \{GossipMessage.src\}$ 
22: end if

```

Through the ontology matching algorithm, each message has the potential to create new associations between concepts from different ontologies. This progressive aspect of ontology matching results in the following network behavior: 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 [18] is used. The algorithm maintains the fixed size of the node view, denoted by $\text{Parameter}_{\text{NodeView}}$, while constantly updating it with a small number of new identifiers (lines 17-22).

The benefits of the gossip execution are twofold: first, the ontology matching component in each node can derive semantic similarity relations between its stored concepts and those received, and 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 distributed ontology matching and semantic discovery can take place.

3.3.4 Optimized Gossip

An important decision in the design of OntoMobil was the separation between the gossip and the routing protocol. While this separation places mobility adaptation to the routing layer and semantic discovery in the application layer, the transmission of concepts many hops away can increase routing overhead and failed transmissions. We have devised and implemented an optimization for the gossip protocol that works with a proactive routing protocol, specifically OLSR, and combines the use of topology-aware membership views with a weighted destination selection mechanism. The idea is to have the node view maintain, in addition to the set of node identifiers, an estimation of the number of hops required to reach each of the corresponding nodes. This information can be recorded each time a gossip message is received and maintained together with an associated timer to track the accuracy of the hop distance. Having a node view that is populated with a distance metric can facilitate an algorithm that does not select identifiers uniformly at random but is biased to those identifiers with a short hop count and timely information. Specifically, in the optimized gossip version, the target of a gossip transmission is not selected based on uniform random sampling but using a weighted random sampling. The node view for node n_i now contains entries of the form $\mathcal{V}_i^N = \{(n_k, w_k), \dots\}$, where n_k is a node identifier, and $w_k = (h \times \Delta t)^{-1}$ is the weight for node k that was h hops away from node i when it was inserted in the node view of node i some Δt time units ago.

3.4 Discovery

Discovery queries in OntoMobil [22] accept a set of concepts describing the capabilities of services or generally the semantics of required content. The mechanism that disseminates these queries exploits the randomized overlay to locate nodes having ontologies with concepts equivalent to the concepts that compose the discovery queries. The specification in this section covers this mechanism and assumes that a matchmaking process at each destination node will identify services or data that are semantically similar to those requested and transmit the reply to the source node.

Some common definitions are provided below followed by the protocol description:

- A node initiating a discovery request becomes the *source* node of the request.
- A request is composed from a set of concepts. The following notation is used:

$$Q = \{c, \dots | c \in \mathcal{V}_i^0, i \in \mathcal{N}\}.$$

- Matches are stored in $\mathcal{R} = \{(c, \{n_i, \dots\}), \dots | c \in Q\}$, where n_i represents a node that has an ontology containing a concept that matches c .
- To simplify the description of the random walk, we define the function $f_S(c)$, where c is a concept with $c \in S$ and S can be either of the two views, i.e., \mathcal{V}^0 , \mathcal{V}^C , or \mathcal{R} . This function returns the set of node identifiers of all matched concepts currently embedded in c . In other words, it returns the values obtained from the set of `om:matchesConceptIn-Node` predicates.

Listing 3. Semantic Discovery

```

1: //RequestMessage represents the semantic query
2: At source node j
3: for all c in Q do
4:   if c in V_j^0 then
5:     R ← f_{V_j^0}(c)
6:   end if
7: end for
8: RequestMessage.ttl ← Parameter_RequestTTL
9: Choose a random nid from the 1-hop neighbors of j
10: send(nid, RequestMessage)
11: At each node receiving RequestMessage:
12: for all c in Q do
13:   if c in V^C ∪ V^0 then
14:     R ← R ∪ f_{V^0}(c) or R ← R ∪ f_{V^C}(c)
15:   end if
16: end for
17: if RequestMessage.ttl = 0 or all concepts found then
18:   D = ∩_{c in Q} (f_R(c))
19:   for all nid in D do
20:     //redirect query to node nid
21:     forward(nid, RequestMessage)
22:   end for
23: else
24:   RequestMessage.ttl ← RequestMessage.ttl - 1
25:   Choose a random nid from 1-hop neighbours
26:   send(nid, RequestMessage)
27: end if

```

The basic discovery mechanism in OntoMobil is flexible and can support many possible variations. These variations are conditioned on several choices, e.g., whether the concepts composing the query exist in the source node's ontology or the nature of the operators that form the query condition. The current variation, manifested in the definition of \mathcal{Q} , is selected because of its simplicity and because it utilizes the mechanisms of concept replication and matching. It does so by allowing the composition of queries from nonlocal concepts, i.e., concepts that do not exist in the ontology view of the source node. The difference being that when a query is solely composed of concepts that exist in the source node's ontology, extracting matching concepts can be accomplished either by only polling periodically the local concepts for new matching associations or by using polling followed by a discovery request. The drawback with a polling-only mechanism is that it becomes overly dependent on the progress of the semantic matching. Combing polling with a discovery request can increase the chances of identifying new associations in other nodes. When a query is composed of nonlocal concepts, a discovery query is mandatory since the query concepts must first be discovered in the network for any subsequently matching associations to be extracted. The motivation to compose queries with nonlocal concepts stems from the ability to create more complex queries by combining concepts in the node's ontology with concepts in the network. Furthermore, it allows the discovery of content using predefined queries by nodes that can only hold a minimal ontology but want to avail of all network knowledge.

We briefly describe the discovery protocol by observing the random walk as traversing a graph built by considering each node as a vertex with outgoing edges to the nodes found in its one-hop neighborhood (lines 1-10). The random walk mechanism is used to collect the identifiers of nodes that maintain ontologies with concepts that match the concepts in the discovery query. The node identifiers constitute addresses of potential content providers since a matching relation by definition indicates partially compatible ontologies. When a query condition is satisfied, i.e., all query concepts are located, or when the query's ttl reaches zero (line 17), the second phase is initiated in which the query is forwarded to the nodes discovered during the first phase (line 21).

4 STOCHASTIC ANALYSIS OF ONTOMOBIL

When considering the applicability of OntoMobil, the size and variability of the concept view become important factors. These factors influence the scalability of the gossip protocol as memory consumption, processing overhead, and the probabilistic guarantees for discovery depend on the distribution of replicated concepts across nodes.

This section uses a stochastic analysis to derive a predictive model of the proposed gossip protocol, given the total number of concepts, the number of nodes, and the characteristic parameters of F_n , F_c , T_t , and T_a . The main aim of the analysis is to formulate a probability measure of the concept view size, which will be used as the foundation for the probabilistic guarantees of the random walk discovery protocol.

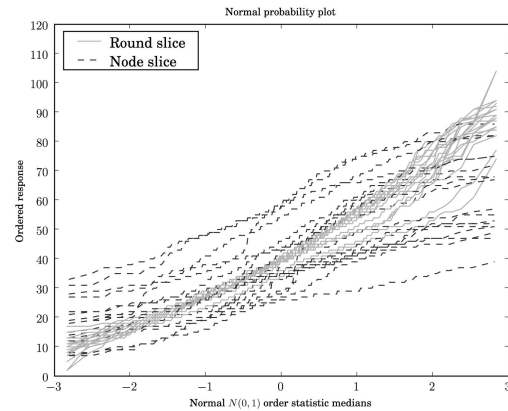


Fig. 1. The variability in the size of the concept view can be measured in space (node slice) or in time (round slice). Data in this figure are taken from simulations with 300 nodes for 300 rounds. Despite the dispersion, it can be seen that both distributions are convergent and approximate the Gaussian. Experiment invariants: $F_c = 4$, $F_n = 2$, $T_t = 2$, and $T_a = 2$.

The variability of the concept view size is modeled using the random variable V . Based on V , the probability mass function (pmf) $f_V(v)$ is derived, which contains complete information about the distribution of V to satisfy the main goal of the analysis.

4.1 Analytical Assumptions

To make analytical modeling tractable, a number of assumptions and simplifications are made. Similar to the system assumptions in Section 3.2, the analysis considers a fixed set of \mathcal{N} participants with cardinality $N = |\mathcal{N}|$, which maintain the same number of concepts $G = |\mathcal{V}^O|$ in their ontology views. This brings the total number of concepts to $G_{total} = N \cdot G$.¹

Recall that concepts are inserted in the concept view if they do not already exist in the two views \mathcal{V}^C and \mathcal{V}^O . Concepts are also removed when they are selected for transmission from the concept view and their age parameter reaches T_a . The analysis presented here focuses on modeling the variability of the concept view size, based on the probability that a certain number of concepts are inserted and removed in each round.

As is evident from the protocol specification, the size of the concept view varies both across nodes and across rounds. Specifically, a node's concept view varies across rounds, while in a certain round, there is variability in the view sizes of the different nodes. We assume that the size distribution across both of these dimensions is Gaussian, and as the number of rounds and the number of nodes increase, the mean and variance of these respective distributions converge. Fig. 1 illustrates the distribution of the concept view size through a normal probability plot taken from a simulation of the gossip protocol with 300 nodes running for 300 rounds. Despite the dispersion between the two data slices and the short-tailed lines, the two slices do approximate the normal distribution, which is sufficient for the purposes of this assumption.

1. The Appendix provides a concise table of the symbols used during the analysis.

The stochastic model provides an analytical distribution that approximates both experimental distributions as derived from the per-node and the per-round samples. This implies that the distribution of the concept view size for a single node across time can also be used as an indicator for the distribution of the concept view sizes across all nodes during a single round. Therefore, the behavior of the gossip protocol is observed from a node that is selected uniformly at random from the nodes in \mathcal{N} .

The four main assumptions that will be used during the analysis are presented as follows:

- A1. *Synchronous gossip transmission.* The stochastic analysis assumes that all nodes transmit in synchronous intervals. At every round, each node in \mathcal{N} transmits a gossip message to F_n other nodes. Node failures or message omissions are not taken into account, and it is assumed that the message latency between all nodes is negligible. It can be inferred that messages sent in a round will be received within the same round. Note that this assumption only requires the upper bound on node-to-node communication to be within the range of the gossip time-out value. This is feasible since there are no real-time requirements and the time-out value can be easily adjusted without influencing protocol correctness.
- A2. *Simple random sample.* All concepts in both the concept and ontology views represent a simple random sample selected from the set of all concepts. This assumption is verified by the agreement between the discovery and analytical results that are presented in Section 4.3. Specifically, this assumption implies that the set of concepts in each node's $\mathcal{V}^C \cup \mathcal{V}^O$ represent a simple random sample from $\bigcup \mathcal{V}_i^O, \forall i \in \mathcal{N}$.
- A3. *Independence of gossip reception and transmission.* The transmission and reception of gossip messages are independent events that happen simultaneously. In practice, transmission and reception of gossip messages have an arbitrary order within a round and occur sequentially. However, treating them as simultaneous events allows the composition of reception and transmission into a single well-ordered gossip action.
- A4. *Sequential insertion of concepts when $F_n > 1$.* The last assumption concerns the reception of multiple gossip messages from different senders, i.e., $F_n > 1$. Section 4.2.2 shows that on the average, $F_c \cdot F_n$ concepts are received by each node in a single round. The assumption of sequential insertion excludes the case where similar concepts may exist in the different F_n transmissions, essentially treating this case similar to the reception of a single gossip message containing $F_c \cdot F_n$ distinct concepts.

4.2 Stochastic Model Specification

The stochastic model considers a network of nodes with empty concept views at the initial round $r = 0$. Let V_r be a random variable representing the concept view size of a randomly chosen node, so that V_r models $|\mathcal{V}^C|$ when $r \geq 0$. From the gossip specification, V_r has range

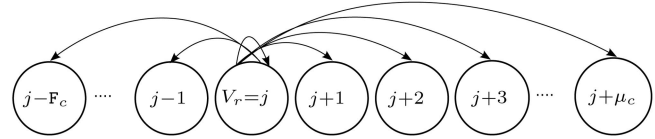


Fig. 2. States and corresponding transitions for a Markov chain that represents the variability in concept view sizes.

$\mathcal{E}_V = \{0, 1, \dots, G_c\}$, where $G_c = G_{total} - G$. The sequence of random variables $\{V_r\}_{r \geq 0}$ can now be considered as a discrete-time and finite-space Markov chain with \mathcal{E}_V as its state space. We now proceed to calculate the state vector and transition matrix of V_r .

4.2.1 State Probability Vector

The state space represents all applicable concept view sizes in unit increments. To represent the state probabilities at round r , we use the vector

$$\mathbf{p}^{(r)} = [p_0^{(r)} \quad \dots \quad p_{G_c}^{(r)}]^T, \quad (1)$$

where $p_i^{(r)} = P[V_r = i]$, $i \in \mathcal{E}_V$, is the probability that the concept view has size i after round r with an initial condition $p_0^{(0)} = P[V_0 = 0] = 1$.

If the stationary vector for $\mathbf{p}^{(r)}$ exists, the state probabilities reach a steady state, and the stationary vector represents the required probability measure of the concept view size. Since the stationary vector can be mapped to the pmf $f_V(v)$, then if it shown that $\mathbf{p}^{(r)}$ has the stationary property, $f_V(v)$ will have been derived.

4.2.2 Transition Probability Matrix

Given that the concept view of a random node has size $V_r = i$ after round r , the transition to size $V_{r+1} = j$ after round $r + 1$ is subject to the following conditions:

- C1. the number of concepts received by the node,
- C2. the number of concepts inserted in the concept view, and
- C3. the number of concepts removed from the concept view because of the node's transmission and the age parameter threshold.

Condition C1 is approximated by computing the mean number of received concepts in each round. Under the assumptions of uniform node views and synchronous gossip transmission A1, each node will receive on the average $\mu_c = F_c \cdot F_n$ concepts from F_n gossip transmissions.

Knowing the mean number of concepts that a node receives in a round, it can be inferred that between two rounds, the size of a concept view ranges from $-F_c$ to μ_c . This variability lies in the interval of the boundary cases where no concepts are inserted and F_c concepts are removed, and all μ_c received concepts are inserted but no concepts are removed.

Fig. 2 illustrates the possible transitions of the concept view size after a round. Each of these transitions can be represented by an event that has zero or more outcomes. Two numbers compose each outcome: the number of concepts inserted in the view and the number of concepts removed from the view. These two numbers correspond to conditions C2 and C3.

For example, the transition to the state of maximum view size reduction occurs when there are zero insertions and F_c concept removals, while moving to the state representing the maximum increase happens when there are zero removals and μ_c concept insertions.

The transition to other states can be computed in a similar way, remembering that most transitions are represented by multiple outcomes. An increase of one in the view size can happen because two concepts are inserted and one is removed or three concepts are inserted and two are removed, etc.

This relationship is revealed if the difference in view sizes between consecutive rounds, i.e., $V_{r+1} - V_r$, is further decomposed into two random variables. Each random variable represents an outcome, with X denoting the number of inserted concepts during a round ($C2$) and Y denoting the number of concepts removed during the same round ($C3$). The two random variables have ranges $\mathcal{E}_X = \{0, 1, \dots, \mu_c\}$ and $\mathcal{E}_Y = \{0, 1, \dots, F_c\}$. It follows that $X - Y = V_{r+1} - V_r$.

As stated in assumption A3, insertion and removal are considered as simultaneous and independent events. Let $P_X(x, i) = P[X = x | V_r = i]$ and $P_Y(y, i) = P[Y = y | V_r = i]$ be used to express the probabilities that x concepts are inserted and y concepts are removed when the concept view size is i . The probability that a view change is $V_{r+1} - V_r$ can then be computed by using the sum of products between $P_X(x, i)$ and $P_Y(y, i)$ for all $x \in \mathcal{E}_X$ and $y \in \mathcal{E}_Y$, where $x - y = V_{r+1} - V_r$.

The view change probability can be generalized to express the transition probability as

$$P_{ij} = P[V_{r+1} = j | V_r = i] = \begin{cases} \sum_{\substack{x \in \mathcal{E}_X, y \in \mathcal{E}_Y: \\ x - y = j - i}} P_X(x, i) P_Y(y, i), & i - F_c \leq j \leq i + \mu_c, \\ 0, & i + \mu_c < j < i - F_c. \end{cases} \quad (2)$$

We can now proceed to compute the values P_{ij} of the probability matrix by deriving $P_X(x, i)$ and $P_Y(y, i)$.

4.2.3 Concept Removal Probability

Let $\zeta(i) = G + i$ be a convenience function that returns the number of concepts in a node's ontology and concept views when the concept view has size i . The concept removal probability is calculated assuming that concepts with different age values are uniformly distributed in the concept view. The probability of removing y concepts, when the concept fanout is F_c can then be calculated as

$$P_Y(y, i) = \frac{\binom{\lceil i/T_a \rceil}{y} \binom{\lfloor \zeta(i) - (i/T_a) \rfloor}{F_c - y}}{\binom{\zeta(i)}{F_c}} \quad y \in \mathcal{E}_Y. \quad (3)$$

Equation (3) expresses the probability of selecting y concepts from the subset of concepts in the concept view that have an age value of $T_a - 1$, while the remaining $F_c - y$ are selected either from the ontology view or from the subset of concepts having an age value different to $T_a - 1$.

4.2.4 Concept Insertion Probability

To calculate the concept insertion probability, it is necessary to identify the number of received concepts that 1) have a ttl

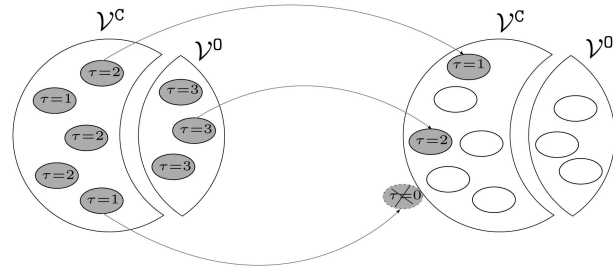


Fig. 3. The distribution of ttl values across concepts in a concept view is proportional to the ttl distribution across concepts in both views of the previous round.

value that is greater than one and 2) do not exist in the concept and ontology views of the receiver. Let $\eta(i)$ be a function that returns the number of concepts in the sender that have a ttl value greater than one and do not exist in the receiver's views.

The probability of inserting x concepts having received μ_c can be calculated as

$$P_X(x, i) = \frac{\binom{\lceil \eta(i) \rceil}{x} \binom{\lfloor \zeta(i) - \eta(i) \rfloor}{\mu_c - x}}{\binom{\zeta(i)}{\mu_c}} \quad x \in \mathcal{E}_X. \quad (4)$$

Equation (4) expresses the probability that x concepts will be selected from $\eta(i)$ concepts and therefore inserted, while the remaining $\mu_c - x$ concepts will either have a ttl value of one or exist in the receiver's views and will accordingly not be inserted.

4.2.5 Computation of $\eta(i)$

As stated in Section 4.2.4, the two factors of the gossip protocol that influence the insertion of received concepts into the concept view are $\mathcal{F}1$) the ttl values of the received concepts and $\mathcal{F}2$) the existence of the received concepts in the concept or ontology views of the receiving node.

In order to compute the probability that the number of concepts selected by the sender will be inserted in the receiver's view, it suffices to relate both $\mathcal{F}1$ and $\mathcal{F}2$ to the view size of the gossip sender. Each of the selected concepts must have a ttl value different to one and cannot exist in the receiver's views. To formulate the concept insertion probability, we require expressions, *predicated* on view size, from which to derive the number of concepts having a certain ttl value and the number of replicated concepts between the sender and the receiver. The following paragraphs outline an analytical derivation of the two expressions.

To compute $\eta(i)$, we must find the distribution of ttl values across the ontology and concept views of a single node for $\mathcal{F}1$ and the distribution of replicated concepts between a sender and a receiver for $\mathcal{F}2$. Since both factors depend on the size of the concept view, their calculation must take place for all concept view sizes.

Knowing that concepts in the ontology view have a ttl value of T_t , the concept view will contain concepts with ttl values in the range of $[1, \dots, T_t - 1]$. Fig. 3 illustrates the intuition that in each round, the number of concepts with a ttl value of τ is proportional to the number of concepts that had a ttl value of $\tau + 1$ in the previous round.

Knowing that the ttl distribution, when the concept view has size i , derives from the ttl distribution of the immediately preceding size $i - 1$, we can use a recurrent function, $g(\tau, i)$, to compute the number of concepts having ttl value τ when the size of the concept view is i . Function $g(\tau, i)$ has the following domain and range:

$$g(\tau, i) : \{1, \dots, T_t\} \times \{0, \dots, G_c\} \rightarrow \{0, \dots, \max\{G, i\}\}$$

and is specified as

$$g(\tau, i) = \begin{cases} 0, & \text{if } \{\tau\}_1^{T_t-1}, i = 0, \\ G, & \text{if } \tau = T_t, \{i\}_0^{G_c}, \\ i \cdot \frac{g(\tau+1, i-1)}{\sum_{\kappa=2}^{T_t} g(\kappa, i-1)}, & \text{if } \{\tau\}_1^{T_t-1}, \{i\}_1^{G_c}. \end{cases} \quad (5)$$

To compute the probability that a transmitted concept exists in the receiver's views (i.e., either the concept or the ontology view), we first need to calculate the number of identical (replicated) concepts in both views between the sender and the receiver. Using an overall set to represent all concepts, under the simple random sample assumption A2, we can use two distinct subsets of equal size selected randomly and with replacement from the overall set to represent the sender's and receiver's concepts.

Let the random variable Z represent the number of replicated concepts between two nodes. The range of Z is $\mathcal{E}_Z = \{0, \dots, \zeta(i)\}$, and its probability distribution can be expressed as

$$P_Z(z, i) = \frac{\binom{\zeta(i)}{z} \binom{G_{total} - \zeta(i)}{\zeta(i) - z}}{\binom{G_{total}}{\zeta(i)}} \quad \text{for } z \in \mathcal{E}_Z. \quad (6)$$

Equation (6) expresses the probability that z concepts are identical when both the sender's and the receiver's views have size $\zeta(i)$. We derive the distribution knowing that $\zeta(i)$ concepts have already been selected from G_{total} for the first subset and formulating the probability distribution of identical concepts based on the second subset. Since the size of the second subset is also $\zeta(i)$, the probability distribution is calculated by having the second subset be composed of z items chosen from the first subset and therefore considered identical, while the rest $\zeta(i) - z$ are selected from the remaining set of all available concepts. Simulation results indicate that the expected value $E[Z]$ is a good approximation to the mean number of identical concepts in both views between the sender and the receiver.

Assuming that the expected number of identical concepts are spread uniformly across the two views in each node, the probability that a transmission contains a certain number of identical concepts can be easily determined.

Factors $\mathcal{F}1$ and $\mathcal{F}2$ can now be expressed with (5) and (6), resulting in the following equation for $\eta(i)$:

$$\eta(i) = \sum_{\tau=2}^{T_t} g(\tau, i) - \left(\frac{\sum_{\tau=2}^{T_t} g(\tau, i)}{\zeta(i)} \cdot E[Z] \right). \quad (7)$$

4.2.6 Stationary Distribution

By construction, the probability matrix derived from (2) is finite. It can be shown [23] that the Markov chain is

irreducible and aperiodic if for some power of the matrix, all its elements are positive.

Each transition matrix constructed under the current model has elements that have either positive or zero values. Each matrix is formulated with a set of well-defined rules, with the following property:

$$\mathbf{P}[ij] = \begin{cases} > 0, & \text{if } i - \beta < j \leq i + \alpha, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

where $\mathbf{P}[ij]$ represents the matrix element at position ij , which has a value calculated from P_{ij} . Specifically, α corresponds to μ_{c_r} and β corresponds to F_c . When $\alpha > 1$ and $\beta > 1$, it is easy to show that for some $r > 1$, the transition probability matrix \mathbf{P}^r will have all its elements positive.

Knowing that the state probability vector $\mathbf{p}^{(r)}$ results in a stationary vector and having formulated the transition matrix, the probability distribution of V is the stationary vector $\lim_{r \rightarrow \infty} \mathbf{p}^{(r)}$, which can be computed from

$$\mathbf{p}^{(r)} = \mathbf{p}^{(r-1)} \mathbf{P}. \quad (9)$$

4.2.7 Discovery Distribution

We can now derive the probabilistic bound on concept discovery for a query composed of nonlocal concepts and specified according to the protocol in Section 3.4:

$$P_{Disc} = 1 - \left(\left(\left(1 - \frac{\overbrace{|\mathcal{V}^C|}^i}{G_{total}} \right) \cdot \left(1 - \frac{\overbrace{|\mathcal{V}^C| + |\mathcal{V}^0|}^{ii}}{G_{total}} \right)^{n-1} \right)^{|\mathcal{Q}|} \right), \quad (10)$$

where $n \geq 1$ represents the number of nodes in the random walk path, including the source node, i represents the probability that a concept exists in the concept view of the source node, ii represents the probability that a concept is found in either of the two views, and $|\mathcal{Q}|$ represents the number of concepts in the query.

4.3 Evaluation of Stochastic Analysis

This section evaluates the stochastic properties of the gossip protocol by studying the accuracy of the analytical model against simulation results. In all experiments, the same parameter values were used for both the simulation and the analysis. We chose two network sizes of 30 and 60 nodes. The varied gossip parameters were node fanout (F_n), age (T_a), and the concept ttl (T_t), while the concept fanout (F_c) and the ontology view size ($|\mathcal{V}^0|$) were kept constant at 4 and 10 concepts, respectively. Section 5 presents more details on the simulation environment.

Fig. 4 demonstrates the results between the analysis and ns2 simulations. For simulation results, the 95 percent confidence interval is calculated over 350 rounds of protocol execution. In each experiment, all nodes record the size of their concept view after every transmission. To illustrate results closer to the steady-state distribution of the concept view, the initial 30 recordings are discarded. Each simulation point in the graph represents the median value from a sequence of average concept view sizes obtained from five

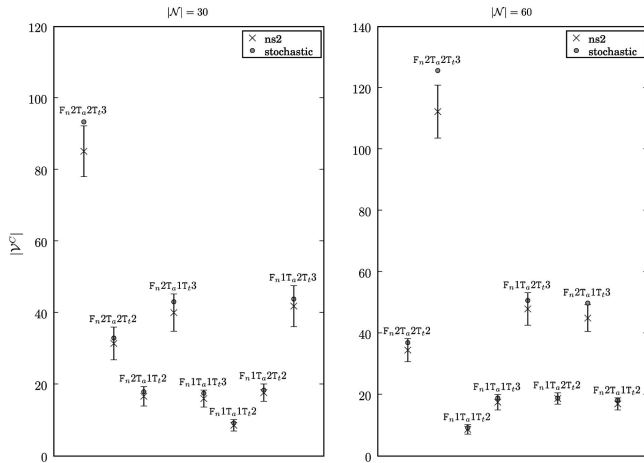


Fig. 4. Comparative results for the size of the concept view between the ns2 simulation and the analysis for a range of the characteristic parameters.

experiments. For the analytical results, what is depicted is the expected value of the state vector ($E[V]$).

For almost all parameter permutation, the stochastic analysis gives a very accurate prediction for the behavior of the gossip protocol. The variability that is observed when the concept view size is large is caused partly by the analytical assumptions made during the derivation of the transition matrix in the analysis and partly by the use of weighted random sampling in the optimized gossip implementation and the failed transmissions in the simulation.

Fig. 5 shows the comparative performance between the ns2 simulation and the analysis of the random walk discovery protocol. In each query, the source node selects one concept ($|Q| = 1$) uniformly at random from a set that includes all available concepts, except the ones in the ontology view of the source node. The source node subsequently searches its concept view for that concept, and if it is not found, the random walk protocol is executed. It is clear that the size of the concept view influences the discovery probability, so we used the gossip ttl ($T_t = 2, 3$) to vary the concept view. After the first 30 rounds of the ns2 simulation, every node transmitted one discovery query per second. In all experiments, the query ttl (ParameterRequestTTL) was set to five. The y -axis depicts the discovery ratio per node in the random walk. It is the probability of locating the query concept during the random walk. The discovery probability is computed using the cumulative number of queries satisfied in each successive node divided by the total number of queries, including those that were lost.

The discovery ratio always increases in proportion to the size of the concept view. With respect to the network size, the discovery ratio shows a linear decrease with an increasing number of nodes and, hence, an increasing number of concepts. Note that this is a natural consequence of the intrinsic relationship in OntoMobil between nodes and content (concepts), which is contrary to the usual treatment of nodes and content as separate entities. OntoMobil can guarantee a probabilistic bound on its properties such as discoverability or the latency of semantic

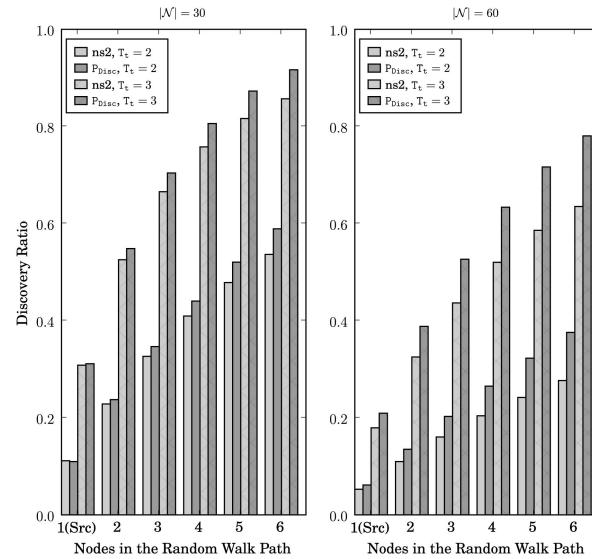


Fig. 5. Comparison of the discovery ratio between the ns2 simulation and the analytical results predicted by (10).

matching, provided that its characteristic parameters adjust to variations in the network size. For example, given a desirable discovery ratio of 60 percent in a network of 30 nodes with the gossip ttl parameter set to two, to guarantee a similar discovery ratio when nodes increase to 60, it suffices to increase the gossip ttl to three. In general, assuming similar ontology view sizes between nodes even as the network grows or shrinks, constant probabilistic bounds on OntoMobil properties are guaranteed by varying the values of the characteristic parameters to keep the ratio of the expected concept view size to total concepts ($E[V]/G_{total}$) constant.

Finally, the correspondence between the analytical and the experimental results verifies the assumption that the union of the concept and ontology views constitute a simple random sample from the population of all concepts (assumption A2 in Section 4.1), which is a further validation of the discovery probability model captured by (10).

5 EXPERIMENTAL EVALUATION

In order to assess the design choices made in OntoMobil, specifically the use of replication and gossiping, we devised *DiscBCast*, a broadcast-based protocol used for the experimental evaluation. DiscBCast does not use replication or gossiping but instead broadcasts a discovery query to all connected nodes. To improve the performance of DiscBCast, we used an established broadcast optimization technique, namely, BCAST, as described in [24]. The aim of DiscBCast is similar to OntoMobil: to discover semantic content in MANETs, assuming multiple heterogeneous ontologies. To facilitate concept matching, a random subset of concepts from the query's source node are always piggybacked with every query. DiscBCast was also implemented in ns2 [25], and simulations between the two protocols were conducted using similar parameters.

The experimental evaluation compared OntoMobil against DiscBCast in terms of the ontology matching latency, the overhead a node sustains due to the distributed

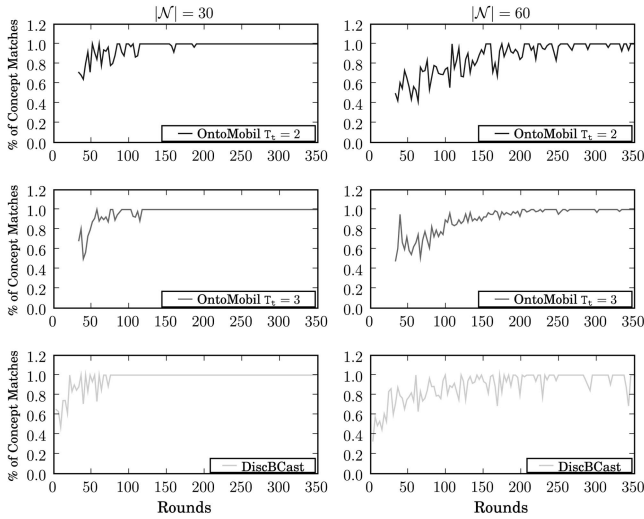


Fig. 6. Comparison of matching latency between OntoMobil and DiscBCast.

matching approach and the network overhead. The latency reveals the required time before a semantic query discovers any potential matches for its concepts. It is measured as the ratio between the matching associations established when the query executes versus all potential matches. The node overhead demonstrates the impact of matching in each node. It is a measure of the number of concept comparisons per second. Finally, the network overhead illustrates the cost of each protocol and is measured in the number of message receptions per second.

5.1 Experimental Setup

Simulations in ns2 were conducted with network sizes of 30 and 60 nodes. Experiments last for 500 seconds each, with each experiment repeated five times. Results are always averages of these five runs. The experimental setup uses the random trip mobility model [26] in a simulation area that maintains a constant density. The area dimensions are 915×915 and $1,297 \times 1,297$ m corresponding to network sizes of 30 and 60 nodes. All nodes act as participants, with each node maintaining an ontology of 10 concepts.

Since DiscBCast does not feature epidemic dissemination or replication, the parameters of node fanout, age, and ttl are not used. Concept fanout is used with similar semantics as in OntoMobil. It represents the number of concepts that are piggybacked with every discovery request.

The fundamental trade-off faced in DiscBCast when setting parameters is the network overhead vis-à-vis matching ratio. Excluding the option to vary the concept fanout parameter, the only two parameters that influence this trade-off is the discovery time-out and the number of nodes querying the network. OntoMobil uses two different time-outs: one for the gossip and one for the discovery protocol. The gossip time-out is selected randomly between the range of $U[0.68, 1.11]$ seconds, while the discovery time-out expires every 1 second. It is clear that a high rate of discovery queries (low time-out) for DiscBCast will accelerate convergence of the multiple ontologies but will also generate excessive traffic. For DiscBCast, we have set the discovery time-out to the value

TABLE 1
Node Overhead Measured in Concept Comparisons per Node per Second

Protocol	$ \mathcal{N} = 30$	$ \mathcal{N} = 60$
OntoMobil $T_t = 2$	400	290
OntoMobil $T_t = 3$	841	890
DiscBCast	445	457

of OntoMobil’s gossip time-out and also used a compensation factor that gives each node a probability of 0.7 of actually executing the discovery query, reducing somewhat the broadcast storm problem [27].

The evaluation of the ontology matching latency used synthetic data. A set of random and unique concepts were generated that populated the ontology view of each node. To establish a notion of similarity, a number of concepts were randomly selected from the ontology view of each node, and a mapping is created between each source concept and one target concept selected from the remainder nodes. The mapping enables the appearance of *transitive mapping* relations, meaning that a concept maybe be related to more than one concept in other ontologies. Note that a concept does not contain a reference to all other concepts in the transitive mapping. Instead, a concept maintains a mapping to only a subset of the concepts that compose the transitive relation. Constructing the mapping relation in such a way is more realistic in a distributed environment, since each ontology is not required to have complete knowledge of all other ontologies. On the other hand, such an approach induces additional latency in order to facilitate complete semantic agreement.

5.2 Results

Fig. 6 depicts the matching latency with respect to protocol rounds. Latency is measured by having each node issue a discovery query every 1 second for OntoMobil and with 70 percent probability once every $U[0.68, 1.11]$ for each node in the DiscBCast simulation. The query is composed of a single concept, and we record the number of established matches in each visited node. As the number of potential matches is known for each concept, it is straightforward to assess the matching ratio. The main factor that influences the number of rounds required before complete semantic matching is established, i.e., when all concepts are augmented with all potential matches, is the network size. The DiscBCast protocol performs marginally better, which is expected because of the higher rate of concept matching facilitated by the broadcast nature of the protocol. As nodes increase, we also observe an increase in the matching latency. This is attributed to the fact that the matching latency increases linearly with the number of total concepts, giving an idealized mean number of concept comparisons per node of $N \times |\mathcal{V}^0|^2$, where $|\mathcal{V}^0|$ is a constant factor. The graph shows that combining a gossip unicast approach with replication can achieve comparable results with a broadcast protocol but, as shown in the next paragraphs, with significantly reduced node and network overhead.

Table 1 displays the overhead incurred by the matching process between OntoMobil and DiscBCast. We measure

TABLE 2
Network Overhead Measured as the Average Number of Received Messages per Node per Second

Protocol	$ \mathcal{N} = 30$	$ \mathcal{N} = 60$
OntoMobil $T_t = 2$	7.7	8.9
OntoMobil $T_t = 3$	6.5	8.2
DiscBCast	34.0	34.1

the overhead using the number of concept comparisons per node per second. It is interesting to observe that there is no significant difference as the number of nodes double. In the case of OntoMobil, this is facilitated by the gossip protocol, which guarantees a constant number of message receptions per node, regardless of the network size. In the case of DiscBCast, the relatively constant number of concept comparisons is due to 1) the increase in the simulation area, guaranteeing a constant node density and hence reducing packet collisions, and 2) the scalability of the BCAST optimization, which prevents network congestion by eliminating redundant transmissions. As already mentioned in Section 4, the determining factor that influences the processing overhead in each node is the size of the concept view. As expected, by increasing the T_t parameter from two to three results in doubling the size of the concept view, which accounts for the higher number of concept comparisons between the two variations of the OntoMobil protocol.

Table 2 shows the network overhead incurred by the two protocols as measured by the number of protocol packets received per node per second. Message reception can provide us with an accurate metric of network saturation, especially when the comparison involves a broadcast and a unicast protocol. We have excluded routing and MAC control packets from this measurement to remove any overhead associated with OLSR and BCAST. As expected, the broadcast nature of DiscBCast incurs a higher toll on the network, leading to increased packet transmission failures and bandwidth saturation.

6 CONCLUSION

This paper presented OntoMobil, a model for the discovery of semantic content that caters for MANETs and semantic decentralization. The model relies on a randomized concept dissemination mechanism to build a semantic overlay topology. Such an overlay facilitates eventual semantic agreement between heterogeneous ontologies and provides a substrate for the discovery of content. We have provided a stochastic analysis of the proposed model and verified the accuracy against simulation results. We have also compared the two main features of OntoMobil, replication and unicast gossiping, against a broadcast protocol in order to shed light on the trade-offs involved. OntoMobil performs favorably in terms of discoverability with significantly reduced overhead.

At the moment, the stochastic analysis (Section 4.2) does not incorporate a failure model. This has an impact in the accuracy of the stochastic model in scenarios with high message failure rate (> 15 percent). Integrating the

probability of lost messages will enhance the predictive power of the model, increasing the accuracy of the expected concept view size.

Another aspect of OntoMobil that requires future attention is the integration of a leave protocol that uses a soft-state mechanism, rather than relying on an explicit disconnection procedure (Section 3.3.2). One approach is to rely on the randomized nature of transmission and use the property that with some probability each node will eventually contact every other node, unless a failure occurs.

APPENDIX

NOTATION

1. \mathcal{V}_i^C is the concept view for node i .
2. \mathcal{V}_i^O is the ontology view for node i with constant size G .
3. \mathcal{V}_i^N is the node view for node i with constant size $\text{Parameter}_{\text{NodeView}}$.
4. T_t is the network-wide constant for the ttl parameter.
5. T_a is the network-wide constant for the age parameter.
6. F_n is the network-wide constant for the node fanout.
7. F_c is the network-wide constant for the concept fanout.
8. G_{total} is the total number of concepts in the network.
9. G is the network-wide constant for the size of the ontology view.
10. G_c is the number of concepts in the network excluding G concepts from the ontology view of a single node.
11. $\zeta(i)$ is a function returning the total number of concepts in both ontology and concept views, given i concepts in the concept view of a node.
12. $\eta(i)$ is a function returning the number of concepts that have a ttl value different to one and do not exist in either the receiver's ontology or concept views, given i concepts in the concept view.
13. \mathcal{N} is the number of nodes in the network.
14. $\text{Parameter}_{\text{NodeView}}$ is a constant representing the number of node ids in a node view, $\text{Parameter}_{\text{NodeView}} = |\mathcal{V}^N|$.
15. $\text{Parameter}_{\text{RequestTTL}}$ is the network-wide constant representing the number of hops before a semantic query expires.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support of Lero: The Irish Software Engineering Research Centre, funded by Science Foundation Ireland. They would also like to thank Vinny Cahill and Stefan Weber for their valuable comments and support in this work.

REFERENCES

- [1] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, *OWL-S: Semantic Markup for Web Services*, W3C, <http://www.w3.org/Submission/OWL-S>, Nov. 2004.

- [2] 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," *Proc. Ninth Int'l Conf. Database Systems for Advanced Applications (DASFAA '04)*, pp. 25-38, 2004.
- [3] K. Aberer, P. Cudré-Mauroux, and M. Hauswirth, "The Chatty Web: Emergent Semantics through Gossiping," *Proc. 12th Int'l Conf. World Wide Web (WWW '03)*, pp. 197-206, 2003.
- [4] A. Doan, J. Madhavan, P. Domingos, and A. Halevy, "Learning to Map between Ontologies on the Semantic Web," *Proc. 11th Int'l Conf. World Wide Web (WWW '02)*, pp. 662-673, 2002.
- [5] S. Castano, A. Ferrara, and S. Montanelli, "H-MATCH: An Algorithm for Dynamically Matching Ontologies in Peer-Based Systems," *Proc. First Int'l Workshop Semantic Web and Databases (SWDB '03)*, pp. 231-250, 2003.
- [6] K. Sycara, S. Widoff, M. Klusch, and J. Lu, "LARKS: Dynamic Matchmaking among Heterogeneous Software Agents in Cyberspace," *Autonomous Agents and Multi-Agent Systems*, vol. 5, no. 2, pp. 173-203, June 2002.
- [7] K. Arnold, R. Scheifler, J. Waldo, B. O'Sullivan, and A. Wollrath, *Jini Specification*. Addison Wesley Longman, 1999.
- [8] E. Guttman, C. Perkins, J. Veizades, and M. Day, *Service Location Protocol, Version 2*. IETF, 1999.
- [9] U.C. Kozat and L. Tassiulas, "Service Discovery in Mobile Ad Hoc Networks: An Overall Perspective on Architectural Choices and Network Layer Support Issues," *Ad Hoc Networks*, vol. 2, no. 1, pp. 23-44, 2004.
- [10] F. Sailhan and V. Issarny, "Scalable Service Discovery for MANET," *Proc. Third IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '05)*, pp. 235-244, 2005.
- [11] V. Lenders, M. May, and B. Plattner, "Service Discovery in Mobile Ad Hoc Networks: A Field Theoretic Approach," *Proc. Int'l Symp. World of Wireless, Mobile and Multimedia Networks (WoWMoM '05)*, June 2005.
- [12] 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," *Proc. 11th Int'l Conf. World Wide Web (WWW '02)*, pp. 604-615, 2002.
- [13] P. Haase, R. Siebes, and F. van Harmelen, "Peer Selection in Peer-to-Peer Networks with Semantic Topologies," *Proc. First Int'l Conf. Semantics in a Networked World (ICNSW '04)*, M. Bouzeghoub, ed., pp. 108-125, June 2004.
- [14] A. Loser, S. Staab, and C. Tempich, "Semantic Social Overlay Networks," *IEEE J. Selected Areas in Comm.*, vol. 25, no. 1, pp. 5-14, 2007.
- [15] B.F. Cooper and H. Garcia-Molina, "SIL: A Model for Analyzing Scalable Peer-to-Peer Search Networks," *Computer Networks*, vol. 50, no. 13, pp. 2380-2400, 2006.
- [16] F. Cuenca-Acuna, C. Peery, R. Martin, and T. Nguyen, "PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities," *Proc. 12th IEEE Int'l Symp. High Performance Distributed Computing*, pp. 236-246, 2003.
- [17] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha, "GSD: A Novel Group Based Service Discovery Protocol for MANETs," *Proc. Fourth IEEE Conf. Mobile and Wireless Comm. Networks (MWCN '02)*, Sept. 2002.
- [18] P.T. Eugster, R. Guerraoui, S.B. Handurukande, P. Kouznetsov, and A.M. Kermarrec, "Lightweight Probabilistic Broadcast," *ACM Trans. Computer Systems*, vol. 21, no. 4, pp. 341-374, 2003.
- [19] J. Luo, P.Th. Eugster, and J.-P. Hubaux, "PILOT: Probabilistic Lightweight grOUp communication sysTEM for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 2, pp. 164-179, 2004.
- [20] A.J. Ganesh, A.-M. Kermarrec, and L. Massoulié, "SCAMP: Peer-to-Peer Lightweight Membership Service for Large-Scale Group Communication," *Proc. Third Int'l Workshop Networked Group Comm. (NGC '01)*, Nov. 2001.
- [21] A. Nedos, K. Singh, R. Cunningham, and S. Clarke, "A Gossip Protocol to Support Service Discovery with Heterogeneous Ontologies in MANETs," *Proc. Third IEEE Int'l Conf. Wireless and Mobile Computing, Networking and Comm. (WiMob '07)*, Oct. 2007.
- [22] A. Nedos, K. Singh, and S. Clarke, "Mobile Ad Hoc Services: Semantic Service Discovery in Mobile Ad Hoc Networks," *Proc. Fourth Int'l Conf. Service-Oriented Computing (ICSOC '06)*, A. Dan and W. Lamersdorf, eds., pp. 90-103, <https://www.cs.tcd.ie/publications/tech-reports/reports.07/TCD-CS-2007-21.pdf>, 2006.

- [23] R.D. Yates and D.J. Goodman, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*. John Wiley & Sons, 1999.
- [24] T. Kunz, "Multicasting in Mobile Ad-Hoc Networks: Achieving High Packet Delivery Ratios," *Proc. Conf. Centre for Advanced Studies on Collaborative Research (CASCON '03)*, pp. 156-170, 2003.
- [25] S. McCanne and S. Floyd, *ns, Network Simulator*, 1997.
- [26] S. PalChaudhuri, J.-Y.L. Boudec, and M. Vojnovic, "Perfect Simulations for Random Trip Mobility Models," *Proc. 38th Ann. Symp. Simulation (ANSS '05)*, pp. 72-79, 2005.
- [27] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Wireless Networks*, vol. 8, nos. 2/3, pp. 153-167, 2002.



Andronikos Nedos received the MSc and PhD degrees from Trinity College Dublin. He is a research fellow in the Distributed Systems Group, Department of Computer Science, School of Computer Science and Statistics, Trinity College Dublin. His main research interests include MANET service discovery protocols, group communication, randomized algorithms, and distributed middleware.



Kulpreet Singh studied civil engineering in the Himalayas and studied group communication systems for mobile ad hoc networks during his PhD at Trinity College Dublin. He is currently with the Distributed Systems Group, Department of Computer Science, School of Computer Science and Statistics, Trinity College Dublin. His interests include new networking paradigms for MANETs that question and challenge the assumptions made for the Internet. He has lived on three different continents up till now.



Raymond Cunningham received the BA degree in mathematics and MSc and PhD degrees in computer science from Trinity College Dublin. He is a research fellow in the Distributed Systems Group, Department of Computer Science, School of Computer Science and Statistics, Trinity College Dublin. His research interests cover the area of mobile distributed systems, distributed systems optimization techniques and adaptive middleware. He has published a number of peer-reviewed papers in the areas of ad hoc networking, peer-to-peer, and distributed optimization.



Siobhán Clarke is a senior lecturer and fellow in the Department of Computer Science, School of Computer Science and Statistics, Trinity College Dublin, where she leads the Distributed Systems Group. Her research interests are programming models and middleware for mobile and embedded systems.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.