A Ubiquitous Knowledge-based System to Enable RFID Object Discovery in Smart Environments

Michele Ruta
SisInfLab, the Information Systems Laboratory
Technical University of Bari, Italy
m.ruta@poliba.it

Eugenio Di Sciascio
SisInfLab, the Information Systems Laboratory
Technical University of Bari, Italy
disciascio@poliba.it

Giacomo Piscitelli
SisInfLab, the Information Systems Laboratory
Technical University of Bari, Italy
piscitel@poliba.it

Floriano Scioscia
SisInfLab, the Information Systems Laboratory
Technical University of Bari, Italy
f.scioscia@poliba.it

Abstract

A ubiquitous Knowledge Base (u-KB) is a distributed and decentralized knowledge base where the factual knowledge (i.e. individuals) is scattered between objects disseminated within a given environment with no centralized coordination. Such a vision enables a truly pervasive environment where autonomous objects set up a self-organized discovery architecture. This paper presents an extended framework to set-up u-KBs. An advanced semantic matchmaking makes possible resource discovery based on metadata stored in RFID (Radio Frequency IDentification) tags without pre-stored and fixed repositories and a dissemination protocol allows an on-demand retrieval of suitable resource descriptions directly from tags located on the objects.

Keywords: RFID, 802.11, Pervasive computing, Semantic web, Resource discovery, Knowledge representation systems

Introduction

Pervasive computing aims at a model of human-computer interaction where information as well as computational capabilities are deeply embedded into the environment, that is into everyday objects and/or actions. The goal of pervasive computing is to reduce the amount of user effort and attention required to benefit from computing systems. As opposed to classical paradigms, where a user explicitly engages a single device to perform a specific task, in pervasive computing the user simultaneously interacts with many computational devices, extracting data from objects deployed into the environment during ordinary activities, even not necessarily being aware of what is happening. “Pervasiveness” refers to manifold aspects involving information storage, processing and discovery. Each mobile host in a given area can access information only on micro devices in its communication range. Consequently, approaches based on centralized control and information storage are utterly impractical in such scenarios.

Radio Frequency IDentification (RFID) technology has been mainly applied to e-commerce, with specific reference to supply chain management and asset tracking. Since low-cost tags can be attached to objects unobtrusively, not altering their appearance and functions, RFID makes possible to identify things, to classify and moreover to discover them. Nevertheless, common exploitation of such a technology does not take into account that nowadays tags – with larger memory and on-board sensors – disclose the possibility of building cooperative environments where autonomous objects (not referring to a supervisor for information exchange and store/update) can be discovered, queried and inventoried without requiring a central control and coordination. A proper dissemination protocol may allow to exactly locate suitable descriptions directly on tags attached to products thanks to a capillary diffusion of tag/reader basic information within the environment. Such a vision allows to build an environment where tagged objects and readers constitute a self-organized, autonomous discovery architecture cross-applicable to several different scenarios.

We define a ubiquitous Knowledge Base (**u-KB**) as a distributed knowledge base whose individuals (assertional knowledge) are disseminated on the objects within the environment. The framework we propose here comprises the specification of components and operations of a u-KB, as well as a distributed application-layer protocol for dissemination and discovery of knowledge embedded within RFID tags dipped in a Mobile Ad-hoc NETwork (MANET). RFID readers are considered as cluster-heads with respect to tags in their radio range and they are able to automatically build up multi-hop inter-reader wireless communication when placed in the same area (Ramanathan and Redi, 2002).

IEEE 802.11 (IEEE 802.11, 1999), along with IP and UDP, is adopted as basic network infrastructure. As reference formalism for resource annotation and matchmaking we exploit ontology languages based on Description Logics (DLs)[1] and originally conceived for the Semantic Web effort, particularly DIG (Description Logics Implementation Group) (Bechhofer et al., 2003), which is a more compact equivalent of OWL-DL Web Ontology Language (McGuinness and van Harmelen, 2004). These design choices provide flexibility in architecture adaptation to specific application requirements. Moreover, reuse of Semantic Web standards can simplify the integration of u-KBs in larger information infrastructures.

The feasibility of the approach has been proved by a simulation campaign which has been performed using ns-2 (Network Simulator, 1995) network simulator package. Effectiveness of metadata dissemination and resource discovery protocol architecture were evaluated.

The remaining of the paper is organized as follows: in the next section we present motivation and possible application scenarios corroborating the proposal; relevant related work are reported in the following section and significant theoretical aspects of the proposed approach ensue. The case study section further explains and motivates the paper before presenting experiments methods and results. Finally, conclusion closes the paper.

Motivation

Motivation for this paper stems from our experience in using Knowledge Representation (KR) approaches – based on Description Logics (DL) formalisms – in pervasive computing, and particularly in semantic-enhanced discovery frameworks based on RFID.

In current applications (see Baader et al. (2002) §1.5 for a survey), Knowledge Representation Systems (KRSs) play a role very much like Database Management Systems (DBMSs). Both are used as central repositories where explicit domain knowledge can be inserted and upon which automated inference procedures can be executed in order to extract implicit knowledge. Hence, in traditional KRSs, a KB is seen as a single fixed entity which is immediately available, either in local storage or via a high-throughput network link. This approach is effective only as long as large computing resources and a stable network infrastructure are granted.

A different approach is needed to adapt KR tools and technologies to functional and non-functional requirements of ubiquitous computing applications. They are characterized by:

- user and device mobility;
- dependency on context (applications must adapt the way they support user tasks according to availability of nearby resources which are volatile by definition);
- severe resource limitations of mobile devices affecting processing, storage, link bandwidth and power consumption.

Hence, knowledge-based systems designed for wired networks are hardly adaptable to wireless ones, due to architectural differences and performance issues.

In the same way, the goal of a pervasive knowledge-based system is to embed semantically rich and easily accessible information into the physical world. This strictly requires shareable vocabularies, otherwise the lack of agreement about the meaning of descriptions would impair interactions among involved actors (Vasudevan, 2004).

The integration of a pervasive knowledge-based system with emerging monitoring and sensing technologies is a relevant issue. Wireless semantic sensor networks (Ni et al., 2005) are a challenging technology. Semantic-based sensory data dissemination and query processing are needed to enable advanced solutions for e.g. environmental monitoring, precision agriculture and disaster recovery. The framework proposed here can fit these requirements and also tackles the problem of improving efficiency by means of cross-layer interactions between data dissemination and routing protocols.

It is noteworthy that the approach we present preserves syntactic infrastructure-based identification (so keeping a backward compatibility with legacy applications). Nevertheless it enables new powerful services. An object can be easily and thoroughly described by means of a semantically annotated description stored within the tag it is associated with. Hence an RFID reader, scanning features of a selected thing, enables a further discovery phase based on a well-structured, non-ambiguous, machine readable annotation. In this way, best matching resources can be discovered and returned to the user.

Such an approach provides several benefits with reference to supply chains applications. Information about a product is structured and complete; it accurately follows the product history within the chain being progressively built or updated during the good life cycle. This improves traceability of production and distribution, facilitates sales and post-sale services thanks to an advanced and selective discovery. Traditional approaches, differently from the one we propose here, do not consider items potentially matching with a user request (expressing needs/requirements of the user) and they do not foresee the possibility of suggesting combinations of items in order to satisfy a user need (which is particularly useful in e-commerce contexts). Let us consider the lifecycle of industrial products: manufacturing and quality control could exploit accurate descriptions of raw materials, components and processes; supply chain management could benefit from improved item tracking; the verification of multi-factor service level agreements between commercial partners could be automated. Furthermore, sale depots could benefit from easier inventory management and could provide ubiquitous commerce (u-commerce) capabilities – like those introduced in Watson et al. (2002), Venkataramani and Iyer (2007) and Ruta et al. (2006) – without expensive investments in infrastructure. Finally, smart post-sale services can be provided to purchasers, by integrating knowledge discovery and reasoning capabilities into home and office appliances (Ruta et al., 2007). In addition, asset management is greatly improved in those scenarios where retrieval should be based on relevant object properties and purposes, rather than mere identification codes.

In healthcare applications, equipment, drugs and patients can be thoroughly and formally described and tracked, not only to ensure that appropriate treatments are given, but also to provide decision support in therapy management. This is an evident improvement in quality of service with respect to infrastructures lacking support for formal semantics, such as Pallapa and Das (2007), which limit discovery to syntactic match of hard-coded attributes. The pervasiveness of infrastructure helps to reduce costs and break barriers between patient management in the hospital and at home.

Likewise, in museums, libraries and archaeological sites, smart semantic-based content fruition can be granted to local visitors as well as to remote clients connected through the Internet, leveraging the lightweight infrastructure already deployed for internal inventory and research.

Related work

Pervasive computing calls for a decentralized and collaborative coordination between autonomous mobile hosts. For the above reasons, pervasive knowledge-based systems have to achieve high degrees of autonomic capability, providing transparent access to knowledge sources that may be present in a given area. This is achieved by exploiting a relatively large number of heterogeneous micro devices – for example RFID tags or wireless sensors – each conveying a small amount of useful information. Due to space, power and cost constraints, they are usually endowed with extremely low storage, little or no processing capability and short-range, low-throughput wireless links.

Middleware infrastructures for ubiquitous computing that can be found in literature usually rely on centralized nodes (i.e. centres of computation/forwarding of network information) for management and discovery of information (Chakraborty et al., 2006, Pallapa and Das, 2007, Vazquez and Lopez-de-Ipina, 2007, Toninelli et al., 2008). In particular, RFID technology is currently used as a mere link between physical objects and a “virtual counterpart” (Römer et al., 2004) in the digital world. Tags trivially store an identification code exploited as a key to retrieve relevant properties of the object from an information server, through a networked infrastructure (De et al., 2004).

Decentralized approaches require a more advanced exploitation of RFID tags as information sources. The biggest obstacle toward this approach is seen in the high cost of RFID tags with enough memory. Notwithstanding, the growing demand of RFID solutions allows to expect that passive RFID tags with higher memory capacity will be available at low cost in the next few years (M Ayoub et al., 2009).

Various works (Venkataramani and Iyer, 2007, Ruta et al., 2007) already proposed approaches for the integration of semantic-enhanced EPCglobal RFID into Mobile Ad-hoc NETworks (MANETs). Semantic annotations could be put into RFIDs attached to objects so that tagged goods stored a semantically rich description featuring the product the tag is clung to. This enables objects equipped with RFID tags to describe themselves toward the “rest of the world” in a self-contained fashion. Nevertheless, a fixed central component was still needed for reasoning over a

Knowledge Base. This led to expensive information duplication within the environment: semantic annotations were placed simultaneously on tags and within the KB; moreover, the reasoning engine was a single point of failure.

These problems can be mitigated if a more distributed approach is followed. Novel paradigms are possible enabling a direct interaction with objects by-passing centralized information repositories: such approaches refer to really pervasive environments where users can directly dialog with objects without intermediaries. In this way an advanced matchmaking can be carried out using metadata stored in RFIDs without fixed knowledge bases and inference tasks can be distributed among readers that provide minimal computational capabilities. Efficient peer-to-peer data dissemination in MANETs is a key technological issue to enable such paradigms. In latest years, epidemic protocols, also known as gossip protocols (Eugster et al., 2004), have been receiving significant interest, as they require no network configuration and provide a good trade-off between algorithmic complexity and performance guarantees (Kempe et al., 2004, Voulgaris et al., 2004). Directed diffusion (Intanagonwiwat et al., 2000) and its evolved variants (Yu et al., 2001, Luo et al., 2005) are also important approaches for large MANETs such as wireless sensor networks. Notwithstanding, they are based on stricter assumptions on network configuration and node capabilities, thus fitting well only specific application scenarios, which are outside the scope of our work.

Advantages of semantic-aware approaches in RFID applications are widely acknowledged – see e.g. Weinstein (2005). Traditional services the technology offers could be considerably improved by introducing a semantically rich object description and discovery capabilities, without depending on centralized infrastructures.

Ruta et al., (2007) proposed an RFID EPCglobal standard evolution supporting an advanced object discovery. Protocol modification enables storage within RFID tags of a set of attributes, a compressed semantic annotation along with the EPC (Electronic Product Code) identifier. They are exploited in the further retrieval phase to perform a more accurate discovery.

From KBs to u-KBs

Fig. 1 reports on the conceptualization of knowledge dissemination and discovery in a pervasive context. The u-KB layer provides common access to information embedded into semantic-enhanced EPCglobal RFID tags populating a smart environment. UDP is used for data exchange in the IEEE 802.11 mobile ad-hoc network. Furthermore, other identification and sensing technologies beyond RFID could be adapted to the general framework, through a semantic support microlayer added to standard protocols. Knowledge provided by a u-KB can be used by applications by means of DL-based reasoning services on top of the semantic discovery protocol defined in the u-KB layer. They can be performed either by hosts in the local MANET or by a remote entity through a gateway exposing a high-level interface (e.g. Web Sevices of RPC[2] or REST[3] type) and translating remote methods into operations on the u-KB.

Here we focus on design, definition and experimental evaluation of the components appearing in bold in Fig. 1. Parallel research effort is being spent into adapting reasoning procedures to resource-constrained mobile devices.

Components and operations of a knowledge base are investigated in this section with the aim of showing how they are modified in case of ubiquitous knowledge bases.

KB components

Given a reference problem in a specified domain, a DL knowledge base models two different kinds of knowledge:

- intensional knowledge, or general knowledge about the reference domain;
- extensional knowledge, which specifically refers to the particular problem.

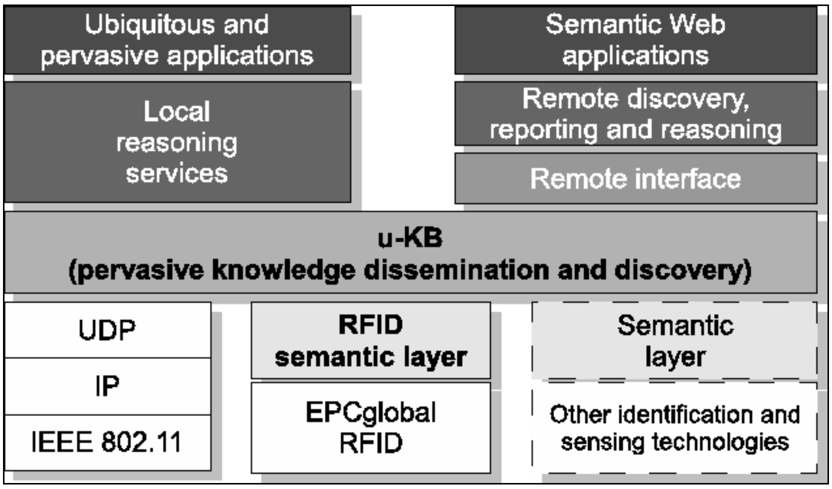

Figure 1. Architecture of the proposed approach

Consequently, a KB has two components (Baader et al. 2002):

- a TBox (terminological box), containing intensional knowledge in the form of an ontology – also called terminology (hence the name) – describing general properties of concepts;
- an ABox (assertion box), containing extensional knowledge – also called assertional knowledge (hence the name) - that is specific to the individuals of the domain.

This hybrid representation approach was first implemented in the early Eighties in the Krypton system (MacGregor, 1991) and it is now followed by most KRSs. It is useful to recall that “intensional knowledge is usually thought not to change – to be ‘timeless’, in a way – and extensional knowledge is usually thought to be contingent, or dependent on a single set of circumstances, and therefore subject to occasional or even constant change” (Baader et al., 2002, p. 17).

KB operations

Current KRSs are characterized in terms of what functions they provide to applications, instead of system data structures and allowed low-level operations (Levesque, 1984). Consequently, designs separate functionality from implementation, leading to both better predictability of system behavior and greater ease of use and integration of a KRS within larger application solutions. Two basic functions were identified for KB management.

- **Tell:** build the TBox and the ABox by means of explicit assertions of terminological knowledge and information about individuals.
- **Ask**: extract (implicit) knowledge from the system. The system answers to queries through inference procedures, determining if the meaning of the query is implied by the information that has been told to the system.

This paradigm has led to detailed and formal interface specifications for Knowledge Representation Systems – such as KRSS (Patel-Schneider and Swartout, 1993) and, more recently, DIG (Bechhofer et al., 2003) – implemented by most KRSs.

The ability to remove information from a KB is also desirable. Most current systems allow to **Un-Tell** (i.e. retract) only information that has been previously told explicitly (see (Brachman et al., 1991) for a discussion). Experience has shown that, for the vast majority of applications, the TBox seldom or never changes after an initial knowledge acquisition phase (see Baader et al. (2002) ch. 8 for a review).

u-KB components

In our approach we preserve the distinction between TBox and ABox.

- The TBox is expressed by means of an ontology document, which can be managed by one or more mobile hosts. For what has been reported above, it can be reasonably assumed that ontologies, defined before object annotation and u-KB deployment, seldom change during normal system activity. Nevertheless, in order to improve the system flexibility, the TBox management could be performed also following paradigms and approaches usually devised in peer-to-peer protocols such as BitTorrent (Cohen, 2008). That is the ontology document file can be entirely available on a single host or it can be fragmented in one or more chunks scattered within the MANET. Notice that since several object classes, described with respect to different ontologies, can co-exist within the same physical space, multiple u-KBs can actually populate the same environment and share the system infrastructure. We adopt unique identifiers like in (Ruta et al., 2006) to unambiguously mark ontologies and to associate each individual to its reference ontology.
- The ABox is scattered within a smart environment, as KB individuals are physically tied to micro devices deployed in the field. In RFID-based scenarios, each individual is a semantically annotated object/product description, stored within the RFID transponder the object is clung to. Each annotation refers to an ontology providing the intensional knowledge for a particular domain. Objects belonging to the same category will be described by means of the same ontology, whereas objects of different categories may refer to different ontologies. In detail, each individual is characterized by:
  - a globally unique item identifier (the EPC – in the case of RFID tags);
  - the OUUID, Ontology Universally Unique IDentifier;
  - a set of data-oriented attributes, which allow to integrate and extend logic-based reasoning services with application-specific and context-aware information processing;
  - a semantic annotation, stored as compressed document fragment in the DL-based language DIG.

u-KB operations

In our u-KB approach, we adhere to the fundamental Tell/Ask paradigm. These operations, however, are implemented in a novel way, coping with the peculiarities of pervasive computing scenarios. Next Section defines in detail data structures and protocol devised to build a u-KB system.

- **Tell/Un-Tell**: these operations are now hidden to users, i.e. no explicit knowledge declaration/retraction is required. Intuitively, a u-KB is formed by knowledge fragments carried by individual micro devices that populate a smart environment in a given time instant and by the ontology they refer to. So simply the presence/absence of them in a given wireless range reveals the information they represent can be added/retracted to/from the KB. In this way, the system allows an autonomic and adaptive knowledge base maintenance. Autonomy relies on the lack of supervision in both information alignment among network hosts and object behavior control. Each node advertises resources it sees in its proximity (e.g. via RFID). Other hosts store advertisements (i.e. packets carrying out information related to resources exposed by nodes) in their cache and forward them to hosts at one-hop distance[4]. The protocol tackles the issue of moving tagged objects entering/leaving the environment. Freshness of advertised individuals is tracked through sequence numbers, so that updates will be automatically propagated. Furthermore, each instance has also a limited time-to-live, so that it will be automatically removed (un-told) from the u-KB if not renewed.

- **Ask**: this operation requires a preliminary resource selection. The requester specifies the ontology identifier and range values for attributes it is interested in. The system then returns IP addresses of hosts owning the ontology chunks and all the individuals that meet the specified criteria[5]. So the requester can reconstruct a local subset of the whole KB, containing only the TBox and individuals which are actually needed. Finally, it is able to submit any Ask-type request to a local or remote reasoning engine.

How to Build a u-KB

Nodes participating to an ad-hoc network do not require any configuration procedure for their inclusion in a MANET. Furthermore each node could act as requester/provider or play the role of a router, forwarding the traffic coming from other hosts in the network.

The proposed framework presents a two-level infrastructure where RFID is exploited at the **field layer** (able to interconnect tags dipped in the environment and readers able to receive the transmitted data), whereas the **discovery layer** is related to the inter-reader ad-hoc communication (see Fig. 2). The communication between the tag field and readers exploits the semantic-enhanced EPCglobal RFID protocol data exchange (Ruta et al., 2007), whereas the data propagation among readers is performed following a data dissemination paradigm in 802.11 (IEEE 802.11, 1999) proposed here. If two nodes are in radio range, the interaction between them can happen directly, whereas it is necessary to adopt a multi-hop routing by exploiting other nodes as intermediate radio links in case of nodes not in direct radio visibility.

A node should not be depending on some other one to advertise/register object descriptions. Resources are autonomously exposed via the enhanced RFID. In the same way, readers should be able to discover them thanks to the preliminary propagation of data each cluster-head has seen in its range. Thus, the resource discovery is based on three stages:

1. the extraction of the object parameters (for carrying object characteristics from field layer to discovery one);
2. resource data dissemination (to make the overall nodes fully aware of the “network content”);
3. the extraction of resource annotations (for carrying semantic-based descriptions from field layer to the discovery one) for further matchmaking. This phase is accomplished after an on-demand request addressed in unicast to a selected node.

Each reader plays a central role in the whole service oriented architecture. It advertises contextual parameters[6] referred to tags in its radio range (at the field layer). During the further phase (at the discovery layer), it will receive requests from nearby nodes (via 802.11). In case, it will extract semantic annotations from tags in range (via RFID EPCglobal) so replying to the requester. A reader maintains a cache containing the advertisements to be matched against requests.

The proposed approach is fully decentralized: address and main characteristics of each resource/tag are autonomously advertised by related readers using small-sized messages throughout the network composed by the other readers.

Care has to be paid in the use of broadcasting mechanism to advertise object features. An uncontrolled exploitation of it could result largely inefficient in terms of bandwidth usage and power consumption (both precious in pervasive environments). In the proposed approach, only resource parameters are advertised in broadcast throughout the network in order to unambiguously identify the location and the category of a resource/tag. Given them, if a node explicitly requires a resource, it will download in unicast the semantically annotated description of it. In this way the advertisement flooding (due to the broadcasting mechanism) is reduced without sacrificing the correct use of a resource.

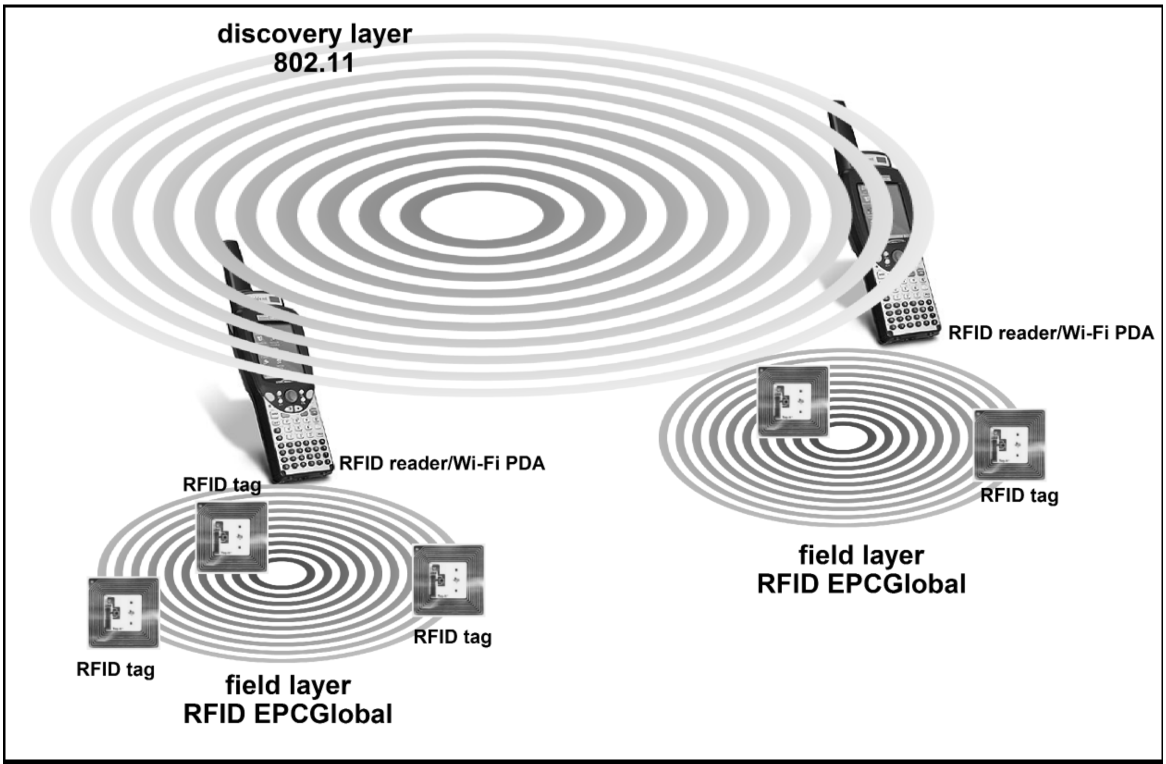

Figure 2. Field and Discovery layers in the proposed framework architecture

As previously described, the proposed framework exploits a controlled diffusion of advertisements containing reader locations followed by an on-demand download of resource descriptions from RFID tags by means of the reader brokerage. In other words, the discovery proceeds in two steps: firstly, a pre-selection of tag population is made taking into account contextual parameters and reference ontologies; furthermore – when (a set of) suitable resources have been identified – semantic annotations are sent in unicast to nodes explicitly interested in them so that they can proceed with the further knowledge-based matchmaking.

We hypothesize each resource in the MANET is labeled by means of the triple *{SOURCE_ADDRESS, OUUID, EPC}*, where the first value is the IP address of the RFID reader which has “seen” the resource, the second one marks the specific reference ontology the resource is associated with, the last one is the Electronic Product Code.

Initially, each reader will advertise, for each resource, the managed reference OUUID as well as some context-aware parameters (i.e. the Time-To-Live (TTL) of the resource). So the initial selection allows to choose only semantically compatible annotations (OUUID matching) with suitable values for context aware parameters.

Before starting whatever reasoning task, the requester has to recompose also the TBox in a consistent fashion. To this aim typical techniques of hybrid P2P sharing are exploited. In particular, a requester attempting to rebuild a TBox will broadcast a specific Protocol Data Unit (PDU) – the *solicit* one, see later on for further details – throughout the network. The packet will contain – among other fields – the OUUID of the reference ontology which will be used by the receiving nodes as hash value to perform a query in its own cache. Ontology chunks will be characterized by a specific OUUID followed by an EPC field in the form *0000 0001-chunk_sequence*. 0000 0001 is a binary header now reserved for future purposes in the EPCglobal standard. It will be followed by the sequence value of the chunk in the overall ontology document starting from 0 and up to the last one. The chunk closing the document will be negative (expressed in a classical mantissa-exponent form) with an abs value for the *chunk_sequence* equal to the last one. In other words, supposing N chunks, the following relation ensues: $chunk\_sequence(N) = -|chunk\_sequence(N-1)|$. A node receiving a solicit for TBox recomposition will reply in unicast with the managed chunk. Only when all the chunks become available, the TBox will be ready for further exploitation. We are currently investigating techniques to split an ontology in a semantic-aware way, by clustering axioms with minimum inter-cluster dependency. The goal is to avoid collecting the whole TBox before reasoning, but only portions that are actually needed by the descriptions of individuals involved in the current request. Since in this paper we aim at assuring on-demand knowledge availability, further details about the reasoning step are not provided.

Summarizing, the discovery procedure occurs in two steps. The first one is syntax-based and aims to select resource descriptions compatible with the request via the OUUID matching and the contextual parameters evaluation. The second one is semantic-based and aims to select only the best available tags. Only in the second step the requester downloads in unicast semantic annotations of resources directly from the provider, so preparing the further matchmaking. This on-demand approach has been chosen considering that semantic descriptions are needed only in the last discovery phase whereas a preliminary ontology-based selection procedure is mandatory. In this way, we obtain a significant reduction of the induced traffic.

Hereafter, previous interaction steps are summarized in a pseudo code sequence. After the advertisement dissemination procedure, when a node has to perform a reasoning procedure the KB must be materialized. Hence, given a request D, expressed with reference to an ontology labeled with the $OUUID_R$ and carrying out a set of contextual parameters C, the requester performs the following steps:

```
1)  Select resources in cache according to both $OUUID_R$ and contextual parameters in C.
2)  Reset current retrieval diameter.
3)  Materialize the TBox $OUUID_R$.
4)  Retrieve semantic annotations of selected resources by sending in unicast request PDUs to resource owners.
5)  Put all the retrieved descriptions in a set R.
6)  Call resource composition algorithm and input R, D, TBox.
7)  Is there an exact solution?
    a) If yes, continue.
    b) If not, go to step 9.
8)  The algorithm outputs the exact solution. Exit.
9)  Is the covering level of a request under a minimum threshold?
    a) If yes, continue.
    b) If not, go to step 14.
10) Increase retrieval diameter.
11) Has maximum retrieval diameter been reached?
    a) If yes, go to step 14.
    b) If not, continue.
12) Broadcast a solicit PDU up to nodes within retrieval diameter. Reached nodes reply with their cache content.
13) Collect incoming descriptions and go to step 4.
14) The algorithm outputs an approximate solution and an explanation on why the solution is non-exact. Exit.
```

Protocol interaction

To perform the data dissemination, resource providers periodically send **advertisement packets** also specifying the maximum hops number for the advertisement travel (MAX_ADV_DIAMETER). The cyclic advertisement diffusion allows to cope with tag and

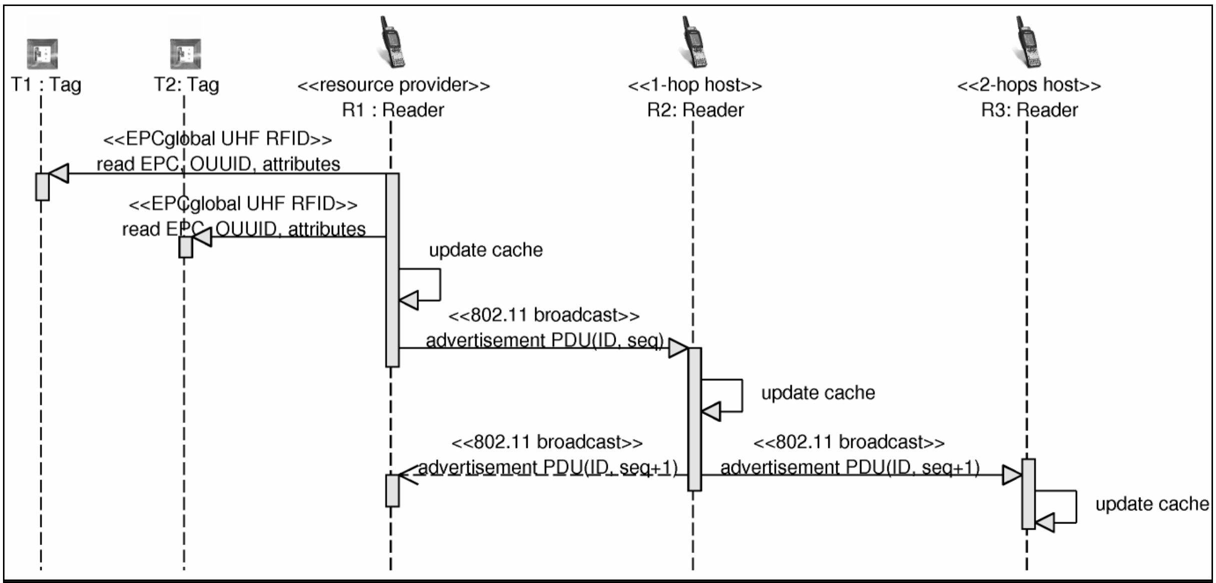

Figure 3. Tag data dissemination phase

reader mobility. As shown in Fig. 3, during their travel, the advertisements are forwarded using MAC broadcasts and can be stored in the cache memories of the nodes they go through.

When starting a resource matchmaking, a node generally attempts to cover the request by using resource descriptions stored within its own cache memory. If some semantic annotations are missing, they can be retrieved in unicast using apposite **request PDUs** whose structure is reported later on. If a requester has no resource descriptions in its cache or if managed resources do not satisfy the request, the node can send a **solicit PDU** with a specified maximum travel diameter (MAX_REQ_DIAMETER) in order to get new resource locators. When receiving a solicit, a node replies (in unicast) providing cache table entries matching parameters contained within the solicit frame. If it does not manage any information satisfying the solicit, it will reply with a “no matches” message. During their travel, replies to request and solicit PDUs are used to update the cache memory of forwarding nodes.

After receiving required information from hosts in its search range, the requester newly performs the matchmaking. If result is still unsatisfactory, the node could try to forward a new solicit request by increasing the maximum search diameter. So these steps can be repeated in an “expanding ring” fashion, until the maximum search diameter is reached. Fig. 4 show the typical sequence and involved actors of the retrieval of semantic annotations.

In the following subsections, the discovery layer will be described in a greater detail. Structure of advertisement, request and solicit PDUs as well as the cache content organization of RFID readers will be explained.

Advertisement PDU

In Fig. 5 the structure of an advertisement PDU is sketched. All the resources inventoried by a reader are advertised by means of a unique advertisement, hence the packet size increases proportionally with the number of tags in the reader range. In what follows we outline PDU fields:

- TYPE: the kind of PDU (see Tab. 1).
- FLAGS: it contains one status flag to distinguish the kind of transmission (uni or broadcast); the remaining flags are reserved for future purposes.
- NUMBER OF RESOURCES: how many tags are in the reader radio range.

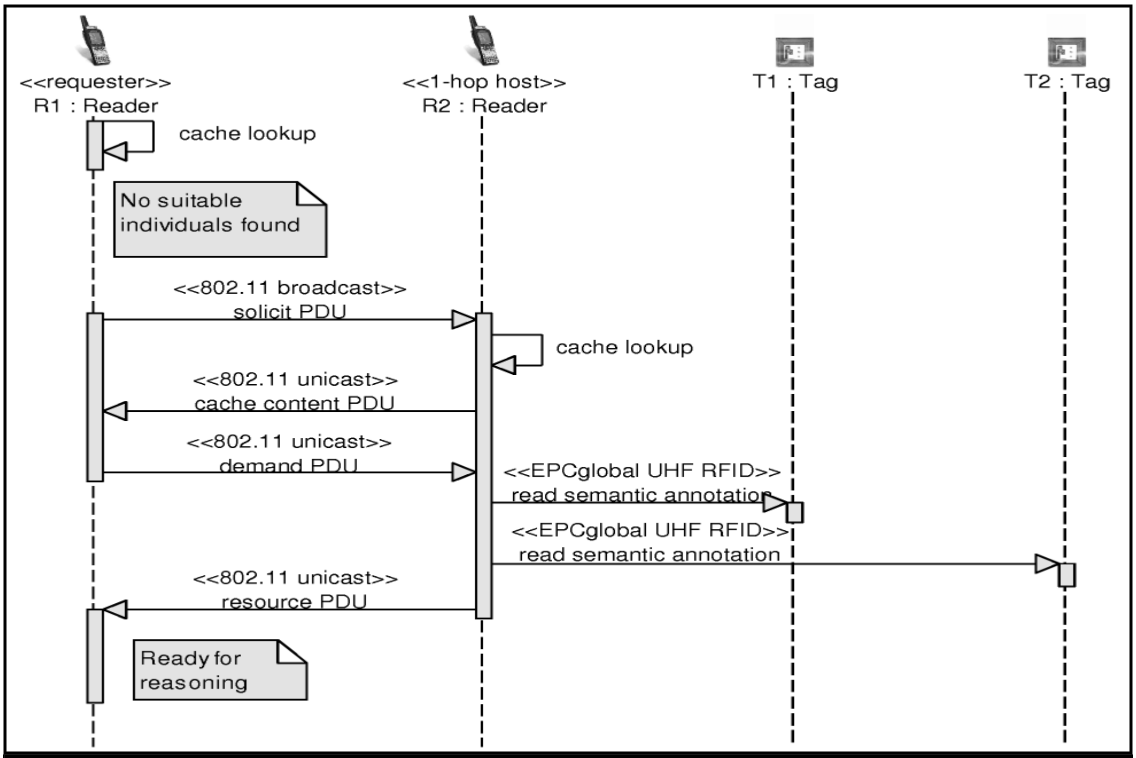

Figure 4. Retrieval of advertised semantic annotations from tags

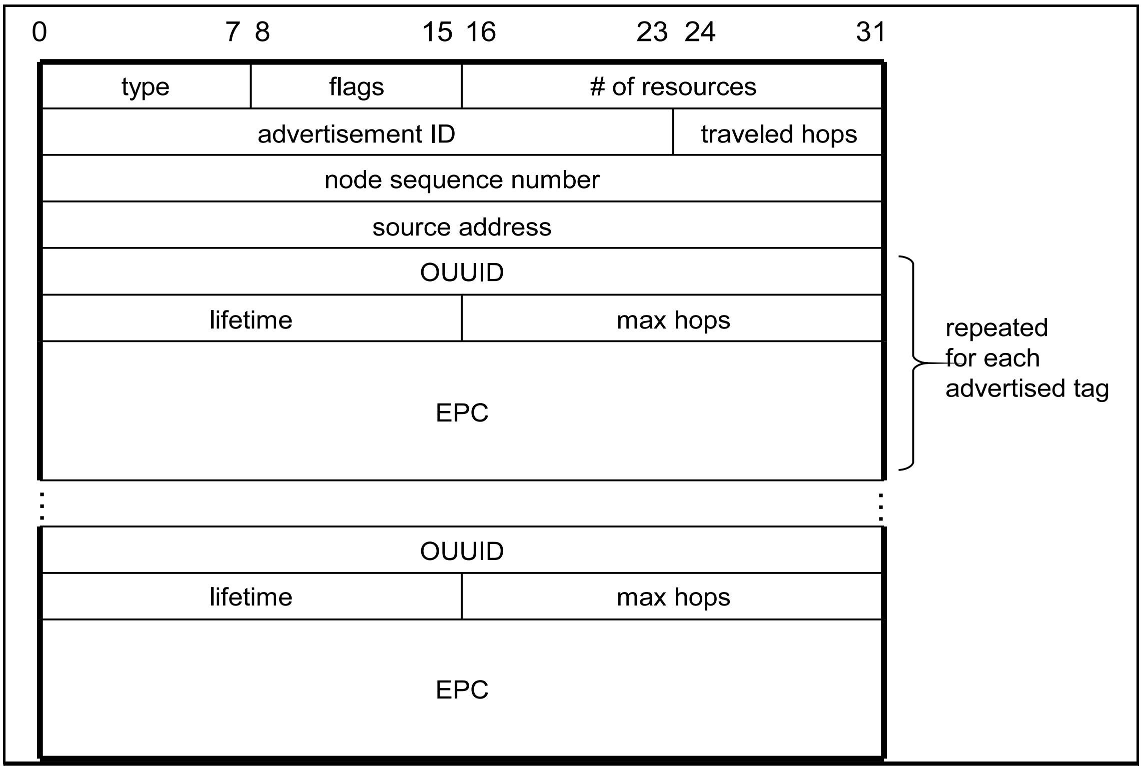

Figure 5. Advertisement PDU

Table 1. PDU types exploited in the proposed framework

Type	Bit Set	PDU
A	0	Advertisement
B	1	Cache entry
C	2	Solicit
D	3	Request
E…L	4…7	reserved

Table 2. Constant values used in the framework

Name	Meaning	Value
DEFAULT_R_TIME	Time interval between two subsequent advertisement transmissions	30000 ms
POLLING_TIME	Time a reader node waits for the echo of the advertisements	7500 ms
MAX_ADV_JITTER	Maximum value for random time waited when forwarding advertisements	600 ms
ONE_HOP_WAIT	Timer set by a requester node after sending a solicit packet, waiting for cache contents reception	2000 ms
HOP_TRAVERSAL_TIME	Time a node needs to process and forward a solicit packet sent by a neighbor	50 ms
ACK_RTT	Timer set by a requester node waiting for ack after a solicit has been sent	50 ms
DISCOVERY_DIAMETER	Current search diameter (in hops) during discovery phase	4
MAX_RETRIES	Maximum number of retransmissions before a reader assumes there are no neighbors	5

- ADVERTISEMENT ID: the reader's sequence number.
- TRAVELED HOPS: the number of hops already traversed by the packet. A reader sets this value to 1 and it is increased every time a node forwards the packet.
- NODE SEQUENCE NUMBER: the sequence number of the node forwarding the packet. If the packet has been sent by a reader this field value coincides with the previous one.
- SOURCE ADDRESS: the IP address of the reader.
- RESOURCE PARAMETERS: a composite, variable length field depending on the number of advertised resources. In particular, it contains the OUUID value, the remaining lifetime of a resource, the maximum hops number for the advertisement travel, and finally the EPC code of the tag.

A reader, which has inventoried a tag series, broadcasts an advertisement every DEFAULT_RTIME milliseconds (see Tab. 2 for constant values). Nearby nodes forward the packet by broadcasting it to their neighbors; as a consequence, the reader listens to the echo of the advertisement packet it originally transmitted. Thus, it can obtain a confirmation of the presence of other nodes in its neighborhood. If the reader does not receive any echo within POLLING_TIME milliseconds (less than DEFAULT_RTIME), it will retransmit the advertisement, assuming that a colli sion or a transmission error has occurred. After MAX_RETRIES attempts, it can be assumed there are no neighbors, so the transmission of the advertisement can be scheduled after a longer timeout in order to reduce power consumption.

When a node receives an advertisement, it extracts information about contained resources and, in case of "new" items, it adds cache entries; otherwise, before updating stored data, the node verifies if the received information is more recent or has ran across a shorter path than the owned one. Particularly, a cache entry is updated if the advertisement carries an ADVERTISEMENT_ID greater or a TRAVELED_HOPS value lower than the ones stored in the cache.

If the cache is updated and the maximum advertisement diameter has not been reached, then the advertisement is forwarded; otherwise the whole packet is silently discarded. This trivial mechanism ensures each mobile node in the network sends the same advertisement at most once. Furthermore, in order to reduce the collision probability (recall that MAC 802.11 protocol does not provide any acknowledgment frame for broadcast transmission), each host waits a random time t $(t \in [0, MAX\_ADV\_JITTER])$ before transmitting. After this check, the node verifies if the maximum advertisement diameter has been reached and, in case, it builds the packet to be forwarded to its neighbors. Firstly, it writes into the new PDU its own sequence number filling the NODE_SEQUENCE_NUMBER field. Then, for each advertised service, it compares the TRAVELED_HOPS value against the MAX_HOPS found in the PDU. If TRAVELED_HOPS = MAX_HOPS the information is discarded, because the maximum advertisement diameter for that service has been reached, otherwise the TRAVELED_HOPS value is increased by 1 and the service information is included into the new packet. In this case, a receiver node increases both traveled hops and node sequence number values (replacing the existing ones).

An analytical model of the data dissemination protocol was developed to estimate whether the inclusion of the 96-bit EPC code in each resource advertisement could lead to an unacceptable network overhead. Let us consider a typical scenario with a partition-free network of n=30 RFID readers and n=30000 tags (1000 tags per reader). Since typical read rates are not above 100 tags/s for EPCglobal Gen-2 UHF RFID (Kawakita and Mistugi, 2006), we reasonably deem that the advertisement period DEFAULT_R_TIME

Table 3. Analytical model of the data dissemination protocol and performance bounds in a typical scenario

Symbol	Meaning (unit)	Formula – empty for input parameters	Value
n	Reader nodes		30
m	Tags		30000
c	$ entry size (B)		50
a	Adv. entry size (B)		20
f	Fixed adv. PDU fields size (B)		16
T_R	R_TIME (s)		30
H	MAX_HOPS		4
S_c	**Needed node $ size (kB)**	$S_c = t\,c$	**1500000**
S_{adv}	Advertisement avg size (B)	$S_{adv} = \frac{t}{n} a + f$	20016
L_{min}	**Min traffic (B/s)**	$L_{\min} = \frac{n\,H}{T_R} S_{adv}$	**80064**
L_{max}	**Max traffic (B/s)**	$L_{\min} = \frac{n^2}{T_R} S_{adv}$	**600480**

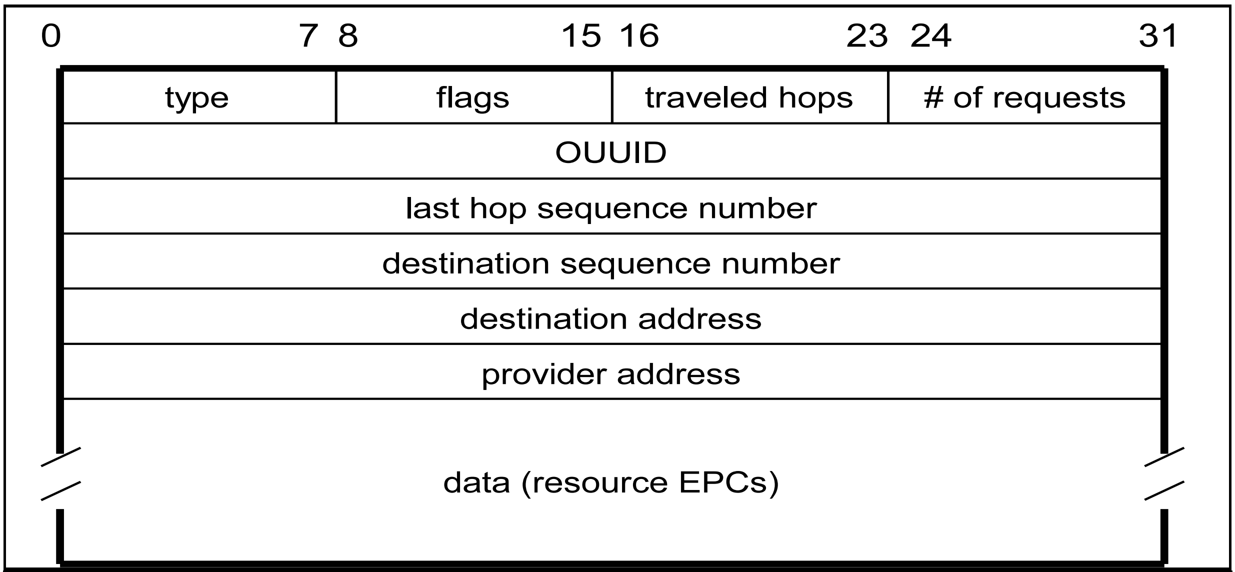

Figure 6. Request PDU

should not be set below 30 s; moreover, let us set MAX_HOPS=4. Tab. 3 summarizes formulas and results of the analytical model. Value of input parameter "*a*" derives from Tab. 5, while "*c*" will be justified in the subsection dedicated to the cache table. As seen above, advertisement PDUs have some mandatory fields followed by a group of fields for each advertised resource. Consequently, the average size S_{adv} of each advertisement is the sum of a fixed and a variable part.

Total traffic generated by data dissemination is characterized by a lower and an upper bound (named L_{min} and L_{max} respectively): the actual value will depend on the topology of the mobile network, particularly on the con nectivity degree (i.e. number of neighbors) of each node. Total traffic is computed as the product of three factors:

- average advertisement size S_{adv}
- overall advertisement frequency, which is n / T_R
- number of propagations for each advertisement: from what has been said previously, it is easy to see that each advertisement is propagated at least MAX_HOPS times and at most n times.

In the above reference scenario, the upper bound is ~590 kB/s, that is ~20 kB/s per reader. Furthermore, it can be noticed that generated traffic is a linear function of tag population size. This grants a theoretically acceptable scalability to the system.

Request PDU

When starting a matchmaking, a node first looks within its cache for entries compatible with the request and, in case, it requires in unicast the corresponding semantic annotations from their respective owners. A request PDU as the one sketched in Fig. 6 is used. In what follows the meaning of introduced PDU fields is summarized.

- TYPE: it is set to 3 (Tab. 1).
- FLAGS: similar to the corresponding field of the advertisement PDU.
- TRAVELED HOPS: how many hops the frame has already gone across.
- NUMBER OF REQUESTS: the number of requested resource descriptions.
- OUUID: unique identifier of the ontology.
- LAST HOP SEQUENCE NUMBER: sequence number of the last node processing the request.
- DESTINATION SEQUENCE NUMBER: the sequence number of the destination node.

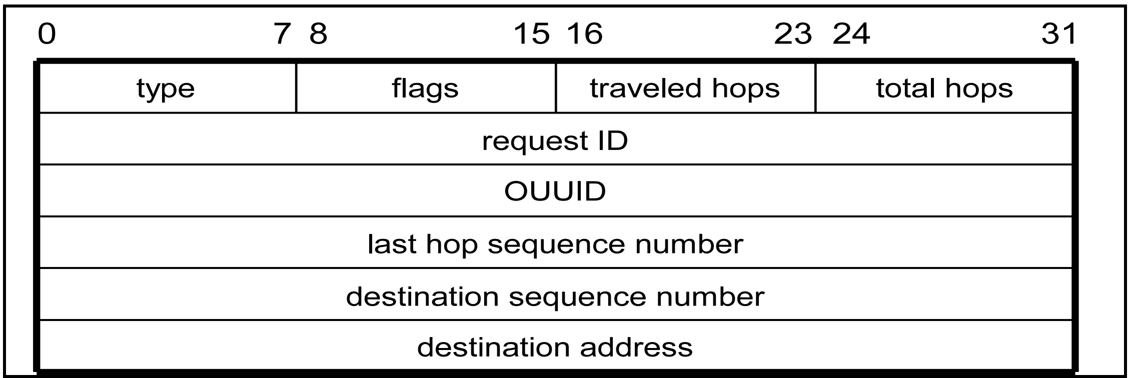

Figure 7. Solicit PDU

- PROVIDER ADDRESS: address of the destination node.
- PROVIDER ADDRESS: address of the destination node.
- DATA: size of this field depends on the number of requests; it contains the EPCs of the tags whose descriptions are required. Each resource ID is twelve byte long, like in the advertisement PDU.

A client requires all missing descriptions at the same time and then waits for replies up to $ONE\_HOP\_WAIT \cdot \mathrm{max\_distance}$ ms (where *max_distance* indicates the maximum hop number between requester node and each provider). When this interval has expired or all requested PDUs have been received, the requester starts the matchmaking.

Solicit PDU

If requester does not own resource descriptions in its cache or if managed resources are

considered insufficient, it should require further descriptions in order to perform a new composition attempt. So it will transmit a solicit packet to nearby nodes.

Basically, the soliciting mechanism is analogous to the advertising one. Fig. 7 shows the format of a solicit packet. Not reported PDU fields are analogous to the previous ones.

- TYPE: it is set to 2 (Tab. 1).
- FLAGS: it maintains the ordinary structure and functionality.
- TRAVELED HOPS: hops the packet has already gone across.
- TOTAL HOPS: total hops the PDU has to skip. Together with the traveled hops field, it regulates the frame travel.
- REQUEST ID: unambiguously labels the PDU in order to distinguish different solicit requests.

A node generating a solicit packet waits for an acknowledgment from each neighbor for ACK_RTT milliseconds. By means of this frame the requester elicits information about nearby nodes: particularly, it can exactly know the number of neighbors. So, before starting the matchmaking or requesting new tag descriptions, it will wait for all the expected cache content PDUs for a time t_w, defined as:

$$t_w = ONE\_HOP\_WAIT + HOP\_TRAVERSAL\_TIME \cdot (current\_hops - 1)$$

where current_hops is the hops number the solicit has to traverse up to reach the DISCOVERY_DIAMETER.

After receiving a solicit PDU, each node located at DISCOVERY_DIAMETER hops from the requester, replies with a cache content PDU in unicast toward the node the solicit came from. Readers receiving a cache content PDU update their own cache and recursively send back a cache content PDU, till the original requester node receives the information it needs. In the following Fig. 8 the soliciting mechanism is portrayed.

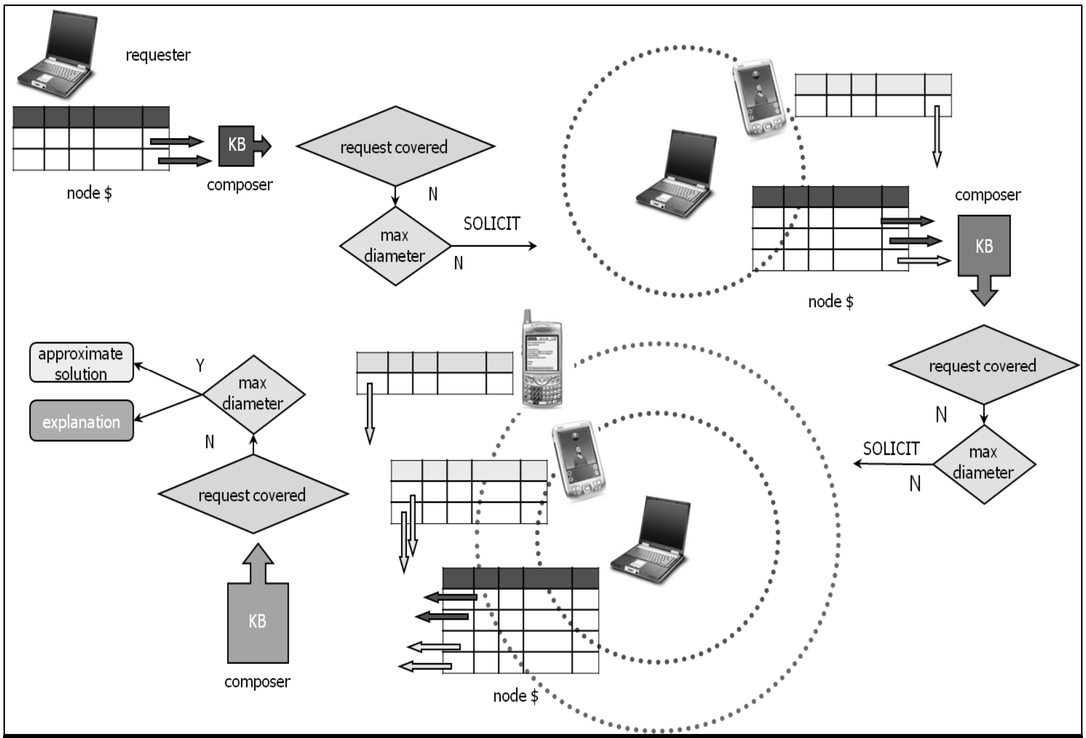

Figure 8. The soliciting mechanism

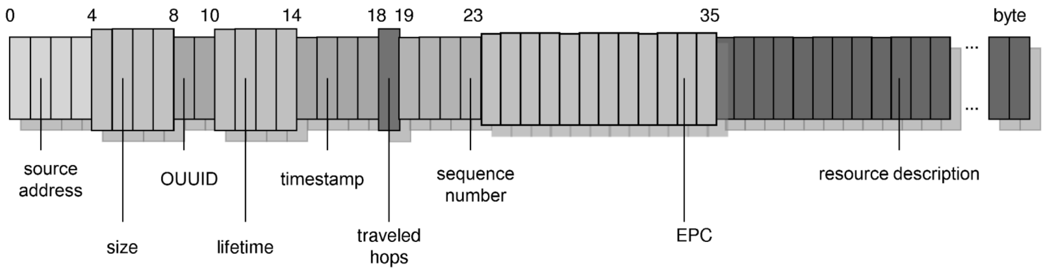

Figure 9. The structure of a record in the reader cache

Cache table management

Each reader manages a cache table where it stores information about both tags in its radio range and tags present somewhere in the network. Fig. 9 shows the structure of a typical entry. In what follows, the content of each field is reported.

- Source address: address of the resource provider.
- Size: annotation size (in byte).
- OUUID: a numeric identifier of the ontology.
- Lifetime: remaining time to live of a resource/tag.
- Timestamp: it marks the last reference to the entry (read/write). That is, when a new resource is stored or when an existing one is invoked, this field is updated.
- Traveled hops: distance (hops number) between provider and cache holder.
- Sequence number: it is referred to the last resource provider.

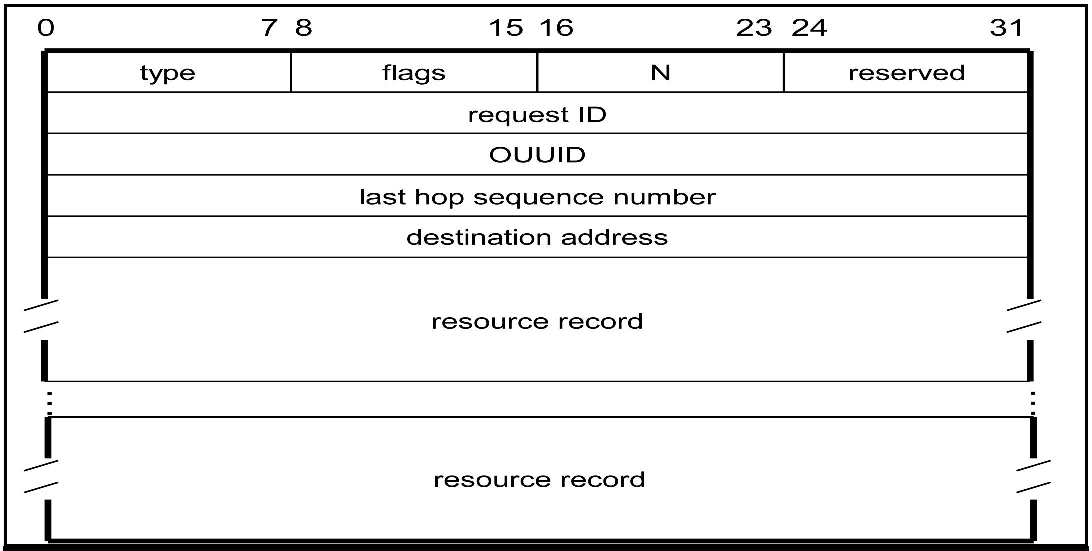

Figure 10. Cache content PDU

- EPC: Electronic Product Code of a given resource/tag.
- Resource description: the semantic annotation of a resource. It will have a variable length, but in some case there could be a pointer to the compressed file containing the DIG description.

An entry could be added to the cache table whenever the node receives an advertisement or a cache content frame arrives.

A cache content packet (whose format is depicted in Fig. 10) has a variable length according to the number of resource handles the PDU transports. Hence a cache update could involve more than one record. PDU fields are outlined hereafter.

- TYPE: it is set to 1 (Tab. 1).
- FLAGS: it maintains the ordinary meaning.
- N: the number of resources handles (and then cache tuples) the packet transports.
- REQUEST ID: identifier of the reference request.
- OUUID: ontology identifier.
- LAST HOP SEQUENCE NUMBER: sequence number of the node sending the packet.
- DESTINATION ADDRESS: requester IP address.

Last fields are the resource records.

Case Study

RFID-based u-KB

The proposed framework has been specifically studied for pervasive computing environments where a wide range of objects/products are endowed with RFID transponders conforming to the EPCglobal standard for class I - second generation UHF tags (Traub et al., 2005). Mobile RFID readers equipped with IEEE 802.11 wireless connectivity (from now on hosts) are responsible for u-KB creation and management. Tagged objects represent KB individuals in our system. Previous works (Ruta et al., 2007) proposed enhancements to the EPCglobal standard which recur in our definition of an individual in a u-KB, as explained above. Here, main features of the semantic-enhanced RFID protocol are briefly recalled, while in the following subsection interactions between RFID technology and the proposed framework will be outlined.

Tag memory is organized in four logical banks (EPCglobal, 2005a):

Table 4. SELECT command able to detect only semantic-enabled tags

Parameter	Target	Action	MemBank	Pointer	Length	Mask
Value	100_2	000_2	01_2	00010101_2	00000010_2	11_2
Description	SL flag	set in case of match, reset otherwise	EPC memory bank	start address	number of bits to compare	bit mask

Table 5. READ command able to extract OUUID from the TID memory bank

Parameter	MemBank	WordPtr	WordCount
Value	10_2	00000010_2	00000010_2
Description	TID memory bank	starting address	read up to 2 words (32 bits)

Table 6. READ command to extract contextual attributes from User memory bank

Parameter	MemBank	WordPtr	WordCount
Value	11_2	00000000_2	00001000_2
Description	User memory bank	starting address	read up to 8 words (16 bytes)

1. *Reserved*, storing optional kill and access passwords;
2. *Electronic Product Code (EPC)*;
3. *Tag IDentification (TID)*, storing tag manufacturer and model identification codes;
4. *User*, optionally present for custom application data.

We exploit two bits (at 15_h-16_h address) in the EPC tag memory area currently reserved for future uses, to indicate respectively whether the tag has a user memory or not and whether it is semantic-enabled or not. Contents of TID memory up to $1F_h$ bit are fixed; optional information could be stored in additional TID memory from 20_h address. There, we store the OUUID of the ontology referred by the semantic annotation in the tag. Finally, contextual attributes are stored within the User memory bank, together with the object semantic annotation expressed in DIG. Due to verbosity of DIG language, the description is compressed using the encoding algorithm proposed in (Scioscia and Ruta, 2009), which allows direct queries over the compressed annotation.

EPCglobal UHF RFID air interface protocol consists in three steps:

1. *Preselection* by a reader of a subset of the tag population currently in range, by means of Select command;
2. I*nventory loop*, where the reader detects one tag in range for each iteration and reads its EPC code;
3. *Access* to the tag memory content.

In our approach, original commands are exploited in novel ways, while keeping full backward compatibility. *Select* command is used to preselect semantic-enabled tags. The inventory step is performed in the standard way. *Read* and *Write* commands in the access step can be used to extract and update the OUUID, the contextual attributes and the semantic annotation of the tagged object.

Interactions with EPCglobal RFID technology

1. Dissemination. After each advertisement period DEFAULT_RTIME, a host scans RFID tags in its range. Only semantic-enabled ones are preselected, by means of a *Select* command with parameters as shown in Tab. 4. Values for the (MemBank, Pointer, Length) triple identify the two-bit memory area starting at 15_h address in the EPC memory bank, which is compared with Mask value 11_2. Since semantic-enabled tags are identified by having both bits set, Target and Action parameters have the effect to set the SL tag status flag only for semantic-enabled tags and clear it for the other ones.

The subsequent inventory step skips tags having SL flag cleared.

EPC codes of semantic-based tags are then individually scanned and TTL of corresponding cache table entries are refreshed. If the host detects a new EPC code not present in its cache table, it will read its OUUID and contextual attributes, exploiting two *Read* commands, as shown in Tab. 5 and Tab. 6 respectively[7].

Data extracted from the RFID tag will be stored in a new cache entry with a fresh sequence number. Conversely, if the EPC code is not detected for an existing local entry in the cache table, the host will wait for the TTL to expire before removing the entry. This prevents the well-known issue of “RFID event flickering[8]” (Römer et al., 2004) from causing incorrect removal/addition of u-KB individuals. At the end of the loop, the cache table is fully updated and the host can issue an *advertisement* PDU to notify individuals to neighboring hosts.

2. Discovery. When a host receives a *request* PDU, it starts an RFID scan of semantic-enabled tags only, as seen above. During inventory, for each detected EPC among those listed in the PDU payload, it reads the compressed semantic annotation stored in the User memory bank of the tag, with a *Read* command as in Tab. 7. Finally, the host replies to the requester. In addition to cache tables described in the previous section, the host may have an optional cache for the most recently used semantic annotations in order to reduce RFID accesses. That may improve response latency and battery life.

3. Ontology provisioning. The EPCglobal Object Naming Service (ONS) (EPCglobal, 2005b) can be used as a fallback mechanism for ontology support in ubiquitous computing contexts if an Internet connection is available. In EPCglobal technology architecture, ONS allows to locate metadata and services associated to a specified EPC. They are provided by the authority managing the object family the tag belongs to. The ONS is based on the Domain Name System adopted to resolve IP addresses corresponding to Internet host names. In particular, ONS format for requests and replies must adhere to DNS standards. Basically the system performs the translation of the EPC code into a domain name and results of this query correspond to valid records of DNS resources. *EPCglobal Network Protocol Parameter Registry*, maintained by *EPCglobal* consortium, includes all the registered service suffixes, such as ws for a Web Service, *epcis* for an EPCglobal Information Service (providing authoritative information about objects associated with an EPC code) and *html* for a Web page of the manufacturer.

Recall that the whole u-KB system is basically structured as a MANET. Hence, if no node in the u-KB manages an ontology *O* (or part of it), an Internet connection is needed in order to retrieve the missing file. If it is endowed with an Internet connection, a node can query the ONS using URI_O derived from OUUID_O and fetch the corresponding compressed ontology file or a chunk of it. For this purpose we use the ONS service and we hypothesize to register within the EPCglobal Network Protocol Parameter Registry the new service suffix dig. The new dig suffix will indicate a service able to retrieve ontologies with a specified OUUID value. Of course the same could be done for OWL or any other ontological language.

In case of EPC derived from the GS1 standard, we assume that the pair of fields used for ONS requests – and referred to the manufacturer and to the merchandise class of the good – will correspond to a given ontology. In fact, that pair exactly identifies the product category. Two goods with the same values for that field parameters will be surely homogeneous or even equal. Nevertheless the vice versa is not always true. Products belonging to the same category, described by means of the same ontology, could have different values for parameters. This is the case of similar goods with different manufacturer or manufactured by the same producer but belonging to different merchandise classes.

4. Implications of RFID data locality. In typical pervasive computing scenarios, the content of a particular RFID tag can be relevant

Table 7. READ command to extract the compressed semantic annotation from User

Parameter	MemBank	WordPtr	WordCount
Value	11_2	00000000_2	00001000_2
Description	User memory bank	starting address	read up the end

to a user only if she is in the same environment as the tagged object itself. For instance, in a large shopping mall, buyers are implicitly interested in products of the department they are currently in. The proposed data dissemination protocol exploits this locality property and delivers advertisements only to hosts within a maximum hop distance from the advertiser (which is supposed to be physically close to the RFID transponder). The value of this operational parameter can be adjusted to balance network load and physical extension of a u-KB, according to application requirements. In the above shopping example, a different u-KB will be built in each store department by IEEE 802.11-enabled RFID readers placed on shelves. Two departments far from each other should not and will not share information. Nevertheless, each department may have a gateway node allowing interactions between the local u-KB system and a back-end warehouse information system.

Experiments

In what follows early evidences about how the proposed framework works are provided. Experimental evaluation has been carried out with the aim of proving: (i) our proposal is coherent with typical wireless network behavior and (ii) introduced enhancements do not impair common-sense duration of resource discovery in wireless ad-hoc contexts. Further analyses have to be performed in non-synthetic real life scenarios in order to show the added value the approach introduces with respect to canonical centralized identification solutions.

Methods

The proposed approach has been verified and tested using ns-2 network simulator in a simulation campaign of the full protocol stack for supporting u-KBs. Evaluated performance metrics are:

1. network load, assessed by means of the total number of packets generated at discovery layer;
2. hit ratio, i.e. percentage of successful resource retrieval, where a hit has to be intended as the delivery of at least three complete resource descriptions referred to the same ontology[9];
3. duration of a complete on-demand knowledge retrieval session.

Furthermore, we assessed the sensitivity of such properties to changes in network topology due to RFID reader and tag mobility.

Several scenarios were created for simulation, each comprising 50 RFID reader nodes moving in a plain 1000 m x 1000 m area. Each simulated host is equipped with a Marvell PXA-255 CPU (Intel Corp., 2003) and an IEEE 802.11 transceiver with omnidirectional antenna, 2 Mb/s nominal bandwidth and 250 m range. Two-way ground signal propagation model (Bakre and Badrunath, 1995) has been adopted.

Host motion follows the random waypoint model (Resta and Santi, 2002) which is characterized by two parameters, *speed S* and *pause time P*. The simulation starts with hosts remaining stationary for *P* seconds, then each host selects a random destination and moves toward it with a fixed speed, randomly chosen in the range [0, S]. After reaching the destination, the host pauses again for *P* seconds, then selects another destination and repeats the previous steps till the end of the simulation, which lasts 900 s. For each scenario, two sets of simulations have been arranged: in the first one S has been varied from 1 to 20 m/s while keeping *P* fixed to 0.01 s; in the second set, the value of *S* has been set to 1 m/s while *P* has been varied from 0 to 900 s.

Scenarios were set up with 3, 5 and 7 hosts detecting RFID tagged resources in their RF-ID range (hereafter *servers*) and 15, 30 and 45 *client hosts*. Resources are present at the beginning of each simulation, whereas requests are generated at randomly chosen instants, uniformly distributed within the simulation interval. Moreover, for each combination of reference parameters (pause time, speed, server and client nodes), 8 simulations were run using different values for the seed of the ns-2 random number generator. Obtained results have been averaged in order to filter out the bias deriving from condi-

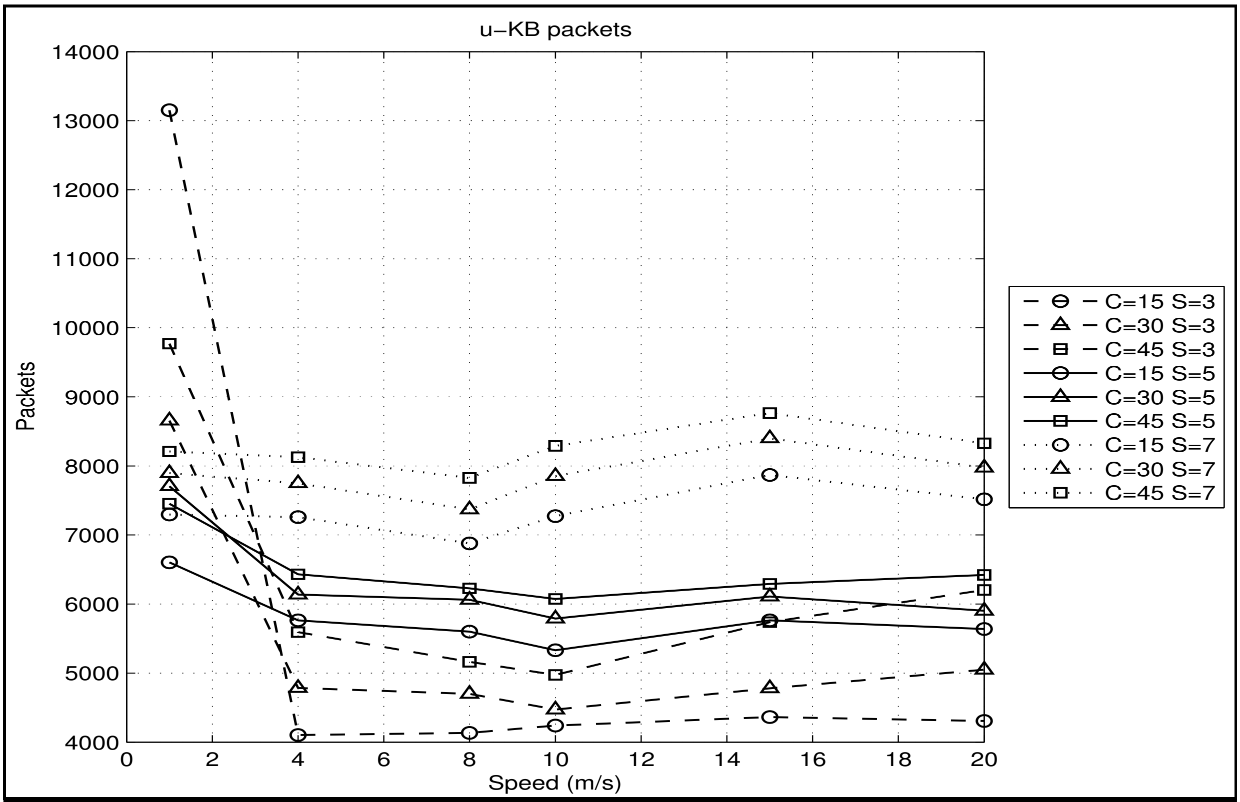

Figure 11. Dissemination and discovery u-KB packets, dependence on the pause time

tions of single scenarios (e.g., high link breakage ratio or network partitions).

Results and discussion

In what follows, outcomes of the overall performance evaluation are presented. Fig. 11 and Fig. 12 report the overall network packets generated by the data dissemination and resource discovery protocol, with dependence from pause time and speed, respectively. Results show that traffic has higher correlation with the number of servers rather than clients. This happens because advertisement packets are regularly sent in a proactive way by servers, even if there are no requests. On the other hand, solicit, cache content and request packets are produced on-demand by client nodes and their neighbors.

Furthermore, Fig. 11 evidences that the number of generated packets decreases when pause time increases. As nodes remain stationary for less time, probability that radio links are lost and advertisement PDUs are not echoed increases: hence advertisers schedule PDUs for retransmission, so increasing the overall traffic.

The hit ratio is depicted in Fig. 13 and Fig. 14, again depending on pause time and speed respectively. Ratio is very high, with values above 90% in all tests and above 95% in more than half the tests.

As a general consideration, this result clearly indicates the relevance of the proposed approach. Analyzing results in more depth, it can be seen that hit ratio increases with the number of resource providers (servers) in the mobile network. This can be explained by two factors. First of all, with more providers the dissemination capillarity increases. Secondly, proposed discovery mechanism induces by itself a load balancing, since the closest servers are contacted first by a requester.

Fig. 15 and Fig. 16 report the time values for a successful resource discovery session, i.e. a complete on-demand knowledge retrieval process including message input/output, cache lookup, and tag access. The maximum

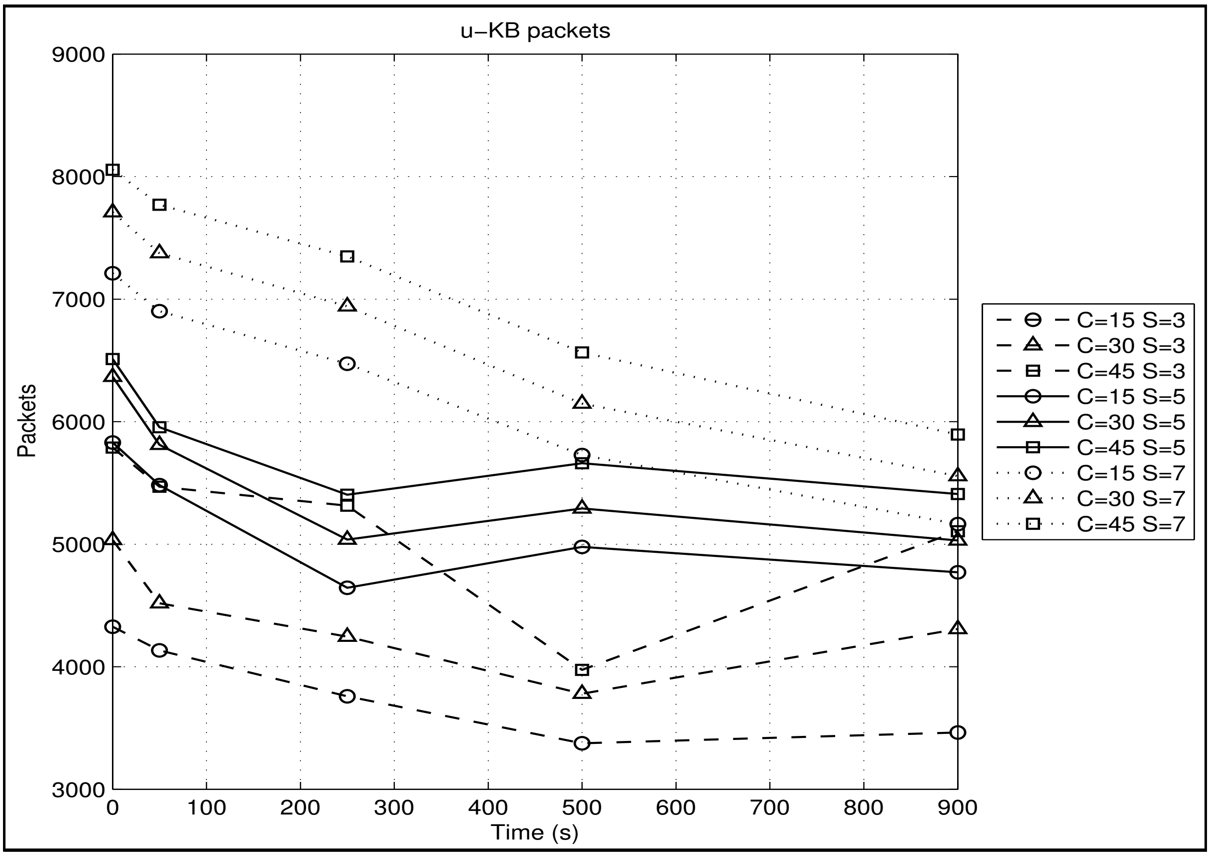

Figure 12. Dissemination and discovery u-KB packets, dependence on the maximum speed

obtained value is 3.1 s, and most of the results are significantly lower.

For a given number of servers, the service time decreases as clients increase. This happens because, when a solicit PDU is answered by cache content PDUs, intermediate nodes cache the resource records. This reduces latency in the reply to later requests.

As a global assessment of the proposed knowledge dissemination and discovery framework, overall network occupancy (in the order of hundreds of kb/s) and latencies (in the order of seconds) are comparable to the ones provided by (De et al., 2004) through analytical models.

It must be noted, however, that the authors' proposal is based on a centralized architecture for ubiquitous RFID resource tracking, which requires complex network configuration and organizational agreements for proper setup. Furthermore, the discovery flexibility is limited with respect to our proposal, since only queries by EPC are allowed and retrieved product information lacks rich and accurate semantics.

Nevertheless, we deem that u-KB dissemination and discovery latencies are still slightly high with respect to requirements of pervasive computing scenarios.

This result may be influenced by the fact that current implementation is not specifically optimized for execution speed. Further techniques should be devised, however, to optimize dissemination and reduce duration of discovery sessions. Then a thorough quantitative performance comparison with other approaches will be possible, in order to assess practical benefits or weaknesses of our proposal.

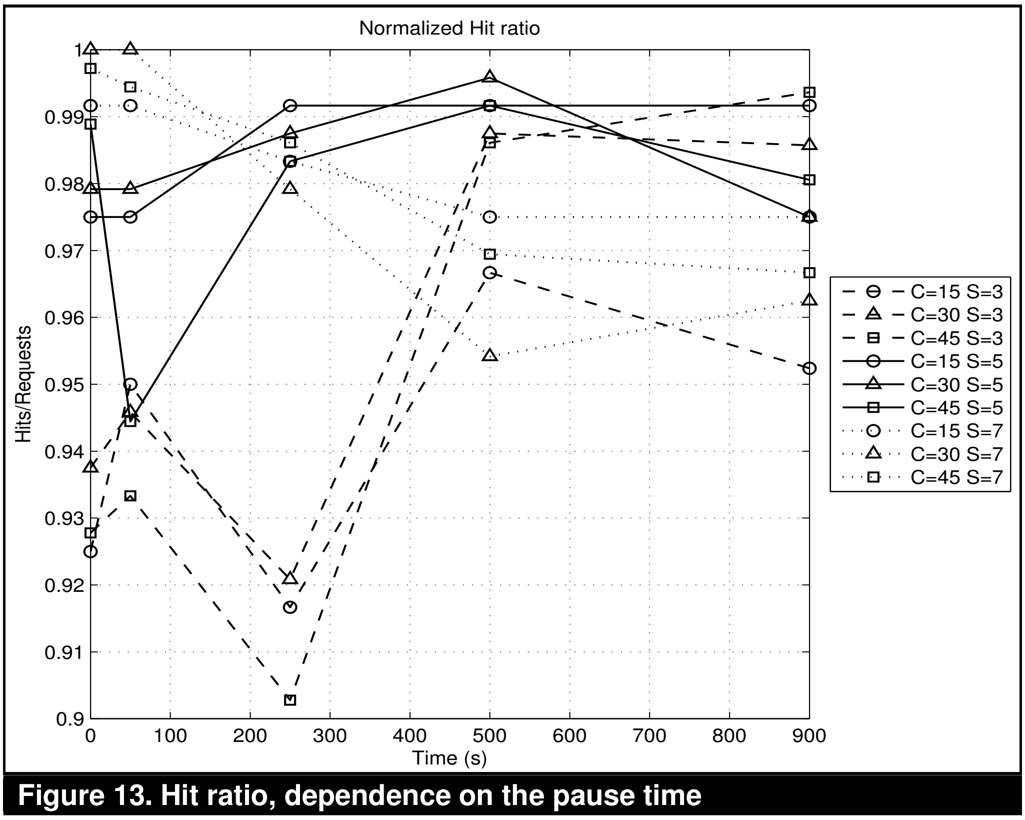

Figure 13. Hit ratio, dependence on the pause time

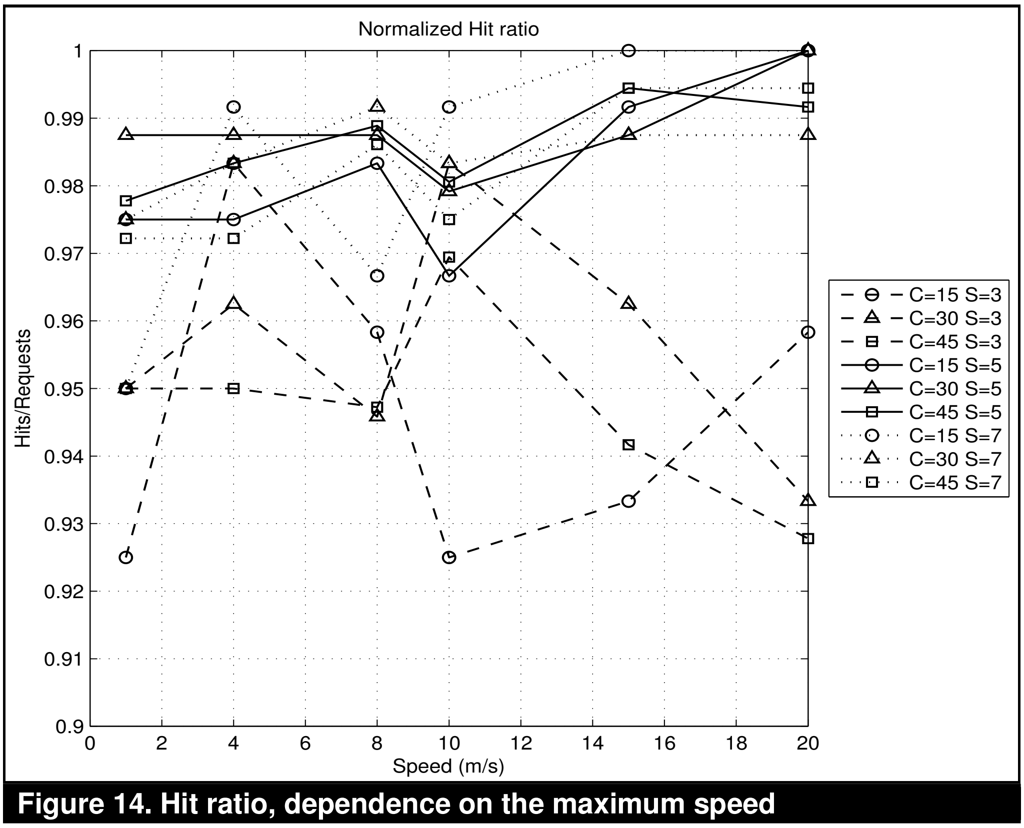

Figure 14. Hit ratio, dependence on the maximum speed

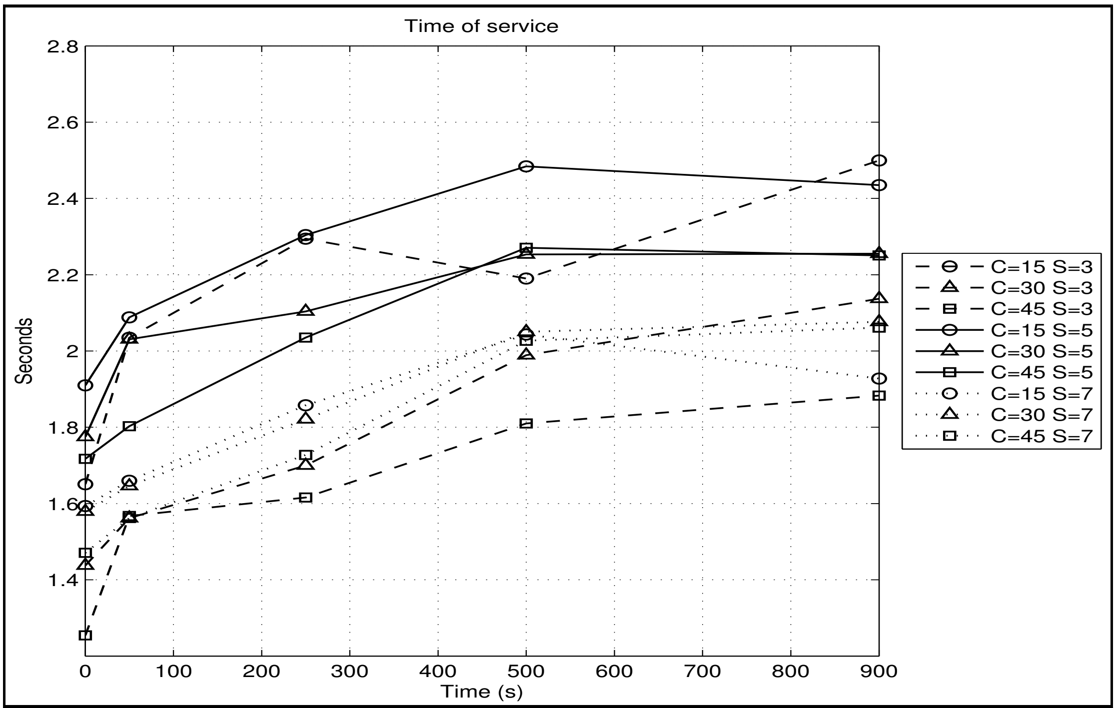

Figure 15. Average time for obtaining a hit, dependence on the pause time

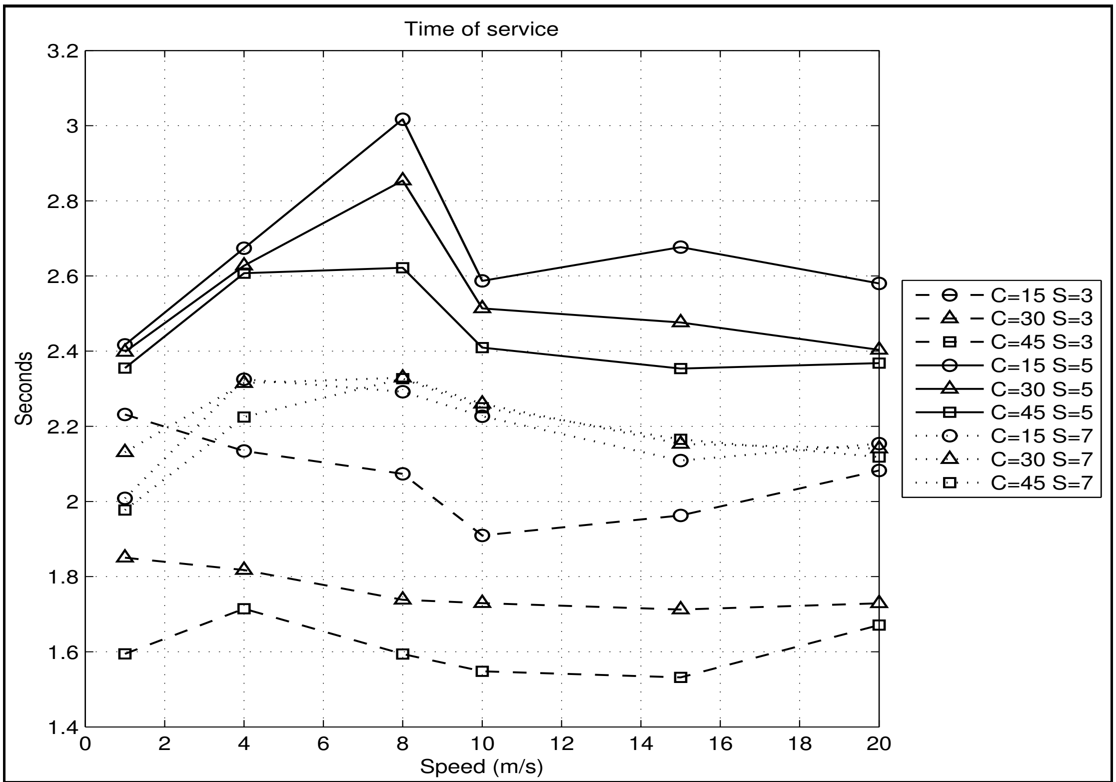

Figure 16. Average time for obtaining a hit, dependence on the maximum speed

Conclusion

We have presented an approach to carry out an advanced object discovery using semantic metadata stored in RFID tags without stable knowledge bases. The proposed framework is supported by a knowledge dissemination protocol, allowing an on-demand retrieval of suitable descriptions directly from tags located on the objects. An analytical model has been developed for a preliminary assessment of requirement of resources. A thorough evaluation of the approach using ns-2 simulation environment has been performed in order to test the feasibility of the proposed approach.

Current solution has now two kinds of limitations. First of all, in the prototype the infrastructure is only simulated, therefore a significant verification cannot be provided yet for performance in the field. A full implementation of the proposed system is ongoing in a testbed with real RFID tags and readers linked among them via 802.11. Secondly, the reconstruction of the TBox component in the knowledge base shows limitations in efficiency and effectiveness; so newer graph-based re-composition methods are now under investigation.

Finally the approach has to be more widely validated through several case studies, in order to expand the scope of the designed framework to different knowledge-based adhoc scenarios.

Future work also includes: wider tests on the proposed methods; extension of the prototype to support multiple KBs in the same environment; improvement of the matchmaking algorithms.

Footnotes

---

[1] DLs are a family of logic formalisms for Knowledge Representation also known as Terminological languages, in a subset of First Order Logic.

[2] Remote Procedure Call

[3] REpresentational State Transfer

[4] Similar mechanisms are employed to disseminate ontology chunks. The system tends toward a steady state where every host is aware of all ontologies and resource instances in the environment. To reduce storage requirements and network load, semantic annotations are not included in advertisements; when needed they are instead provided on-demand.

[5] On-demand provisioning of KB individuals reduces transmissions for data alignment and prevents propagation issues in case of description update or resource removal. Additionally, filtering avoids unnecessary data transfers.

[6] Contextual parameters are numerical values referred either to context or user contingencies and not conveying any semantics. The choice of a set of values depends on the specific applications (e.g. in a mobile commerce context the price of a good is a useful parameter whereas in mobile learning scenarios the lesson duration is more appropriated). They can be properly exploited (with the aid of score combination functions) to improve the practical significance of semantic matchmaking.

[7] *Read* command allows to read one or more 16-bit memory words from any of the four tag memory banks. MemBank parameter identifies the memory bank (as in *Select* command). WordPtr and WordCount are the starting address and the number of memory words to be read, respectively; if WordCount is 0, all the memory words will be read up to the end of the selected bank.

[8] Due to collisions, a tag might not be detected in every consecutive scan. This phenomenon can trigger spurious leave-enter event pairs.

[9] This reference value has been elicited from preliminary tests, using different domain ontologies. Obviously, the greater the number of available resources per node, the greater the probability of success.

Acknowledgments

We wish to acknowledge support of Apulia project PS_121 “Telecommunication Facilities and Wireless Sensor Networks in Emergency Management”.

References

Baader F., Calvanese D., Mc Guinness D., Nardi D., Patel-Schneider P. (2002). The Description Logic Handbook. Cambridge University Press.

Bakre A., Badrunath B.R. (1995). “I-TCP: indirect TCP for mobile hosts,” *Proceedings of the 15th International Confe-*

rence on Distributed Computer Systems, 136-143.

Bechhofer S., Möller R., Crowther P. (2003). "The DIG Description Logic Interface," *Proceedings of the 16th International Workshop on Description Logics (DL'03)*, 81.

Brachman R., McGuinness D., Patel-Schneider P., Resnick L., Borgida A. (1991). "Living with CLASSIC: When and how to use a KL-ONE-like language," *Principles of Semantic Networks*, 401-456, Morgan Kaufmann Publishers.

Chakraborty D., Joshi A., Yesha Y., Finin T. (2006). "Toward distributed service discovery in pervasive computing environments," *IEEE Transactions on Mobile Computing*, 5(2), 97-112.

Cohen B. (2008). "The BitTorrent Protocol Specification," version 11031. Retrieved from: http://www.bittorrent.org/beps/bep_0003.html.

De P., Basu K., Das S. (2004). "An ubiquitous architectural framework and protocol for object tracking using RFID tags," *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS 2004)*, 174-182.

EPCglobal Inc. (2005a). "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz," EPCglobal Ratified specification.

EPCglobal Inc. (2005b). "Object Naming Service (ONS) - ver. 1.0," EPCglobal Ratified specification.

Eugster P.T., Guerraoui R., Kermarrec A.M., Massoulie L. (2004). "Epidemic information dissemination in distributed systems," *Computer*, 37(5), 60-67.

IEEE 802.11 (1999). "Information Technology Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std. 802.11," *ISO/IEC 8802-11*. First edn.

Intel Corp. (2003). "Intel PXA255 Processor Developer's Manual," Retrieved from: http://www.xaiu.edu.cn/general/yxsz/xxy/eelab/book/PXA255_Developer.pdf. Accessed on 17 October 2008.

Intanagonwiwat C., Govindan R., Estrin D. (2000). "Directed diffusion: A scalable and robust communication paradigm for sensor networks," *Proceedings of the 6th annual international conference on Mobile computing and networking*, 56-67.

Kawakita Y., Mistugi J. (2006). "Anti-collision performance of Gen2 air protocol in random error communication link," *Proceedings of the International Symposium on Applications and the Internet Workshops*, 68-71.

Kempe D., Kleinberg J., Demers A. (2004). "Spatial gossip and resource location protocols," *Journal of the ACM (JACM)*, 51(6), 943-967.

Levesque H. (1984). "Foundations of a Functional Approach to Knowledge Representation," *Artificial Intelligence*, 23, 155-212.

Luo H., Ye F., Cheng J., Lu S., Zhang L. (2005). "TTDD: two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, 11(1), 161-175.

M Ayoub K., Manoj S., Brahmanandha P.R. (2009). "A Survey of RFID Tags," *International Journal of Recent Trends in Engineering*, 1(4), 68-71.

MacGregor R. (1991). "The Evolving Technology of Classification-Based Knowledge Representation Systems," *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, 385-400, Morgan Kaufmann Publishers.

McGuinness D.L., van Harmelen F. (2004). "OWL Web Ontology Language," *W3C Recommendation*. Retrieved from: http://www.w3.org/TR/owlfeatures/.

Network Simulator (1995). "The Network Simulator – ns-2," Retrieved from: http://www.isi.edu/nsnam/ns/

Ni L.M., Zhu Y., Ma J., Li M., Luo Q., Liu. Y. et al. (2005). "Semantic Sensor Net: An Extensible Framework," Lecture Notes in Computer Science, 3619, 1144-1153, Springer Berlin / Heidelberg.

Pallapa G., Das S. (2007). "Resource Discovery in Ubiquitous Health Care," *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, 2.

Patel-Schneider P., Swartout B. (1993). "Description-Logic Knowledge Representation System Specification," KRSS Group of the ARPA Knowledge Sharing Effort.

Ramanathan R., Redi J. (2002). "A brief overview of ad hoc networks: Challenges and directions," I*EEE Communications Magazine,* 40, 20-22.

Resta G., Santi P. (2002)." An analysis of the node spatial distribution of the random waypoint model for ad hoc networks," *ACM Workshop on Principles of Mobile Computing (POMC)*, 44-50.

Römer K., Schoch T., Mattern F., Dübendorfer T. (2004). "Smart Identification Frameworks for Ubiquitous Computing Applications," *Wireless Networks*, 10, 689-700.

Ruta M., Di Noia T., Di Sciascio E., Donini F.M. (2006). "Semantic-Enhanced Bluetooth Discovery Protocol for M-Commerce Applications," *International Journal of Web and Grid Services*, 2, 424-452.

Ruta M., Di Noia T., Di Sciascio E., Scioscia F., Piscitelli G. (2007). "If objects could talk: A novel resource discovery approach for pervasive environments," *International Journal of Internet and Protocol Technology*, *Applications and Trends*, 2, 199-217.

Scioscia F., Ruta M. (2009). "Building a Semantic Web of Things: issues and perspectives in information compression," *Proceedings of the 3rd IEEE International Conference on Semantic Computing (ICSC 2009)*, 589-594.

Toninelli A., Corradi A., Montanari R. (2008). "Semantic-based discovery to support mobile context-aware service access," *Computer Communications*, 31(5), 935-949.

Traub K., Allgair G., Barthel H., Bustein L., Garrett J., Hogan B., et al. (2005). "EPCglobal Architecture Framework," EPCglobal Ratified specification.

Vasudevan V. (2004). "Ensembleware: contextual service provisioning in an ubiquitous services world," *International Symposium on Applications and the Internet*, 10.

Vazquez J.I., Lopez-de-Ipina D. (2007). "mRDP: An HTTP-based lightweight semantic discovery protocol," *Computer Networks,* 51(16), 4529-4542.

Venkataramani G., Iyer P. (2007). "Semantics-aware RFID Middleware for Personalized Web Services to Retail Customers," *International Workshop on Service-Oriented Engineering and Optimization*, Goa (India), December 2007.

Voulgaris S., Jelasity M., Van Steen M. (2004). "A robust and scalable peer-to-peer gossiping protocol," *Agents and Peer-to-Peer Computing*, 9-25.

Watson R., Pitt L., Berthon P., Zinkhan G. (2002). "U-Commerce: Expanding the Universe of Marketing," *Journal of the Academy of Marketing Science*, 30, 333-347.

Weinstein R. (2005). "RFID: A technical overview and its application to the enterprise," *IT Professional*, 7, 27-33.

Yu Y., Govindan R., Estrin D. (2001). "Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks," *UCLA Computer Science Department Technical Report*, UCLA-CSD TR-01-0023.

About the Authors

Michele Ruta received the laurea degree in Electronics Engineering from the Technical University of Bari in 2002 and the Ph.D. in Computer Science from Technical University of Bari in 2007. He is currently assistant professor at Technical University of Bari. His research interests include pervasive computing and ubiquitous web, mobile e-commerce applications, knowledge representation systems and applications for wireless ad hoc contexts. On these topics, he has co-authored papers in international journals and conferences. He is involved in various research projects related to his research interests. He co-authored a paper that received the best paper award at the conference ICEC-2007 and has been Program Committee member of several international conferences and workshops in areas related to his research interests.

Eugenio Di Sciascio received the master's degree with honours from University of Bari, and the Ph.D. from Politecnico di Bari (Technical University of Bari). He is currently full professor of Information Technology Engineering at Technical University of Bari, and leads the research group of SisInfLab, the Information Systems Laboratory of Technical University of Bari. Formerly, he has been an assistant professor at University of Lecce and associate professor at Technical University of Bari. His research interests include multimedia information retrieval, knowledge representation and e-commerce. He is involved in several national and European research projects related to his research interests. He co-authored papers that received best paper awards at conferences ICEC-2004, IEEE CEC-EEE-2006 and ICEC-2007.

Giacomo Piscitelli received the master's degree with honours in Physics from University of Bari. He is currently full professor of Information Technology Engineering at Technical University of Bari. His research interests include real time operating systems, information retrieval, knowledge representation and ubiquitous computing. He is involved in several national and European research projects related to his research interests. He co-authored various papers in international journals and conferences.

Floriano Scioscia received the master's degree in Information Technology Engineering from Politecnico di Bari (Technical University of Bari) in 2006, and the Ph.D. in Information Engineering in 2010 from the same institution. He is currently a post-doc research fellow at the Information Systems Laboratory (SisInfLab) in the same University. His research interests include pervasive computing and the Internet of Things, knowledge representation systems and applications for wireless ad-hoc networks and ubiquitous contexts. He co-authored about 30 papers in international edited books, journals and conferences and received the best paper award at the conference ICEC-2007.