Ubiquitous Knowledge-based Framework for RFID Semantic Discovery in Smart U-Commerce Environments

Michele Ruta, Floriano Scioscia, Tommaso Di Noia, Eugenio Di Sciascio, Giacomo Piscitelli
SisInfLab- Politecnico di Bari
Via Re David 200, Bari, BA I-70125, ITALY
{m.ruta, f.scioscia, t.dinoia, disciascio, piscitel}@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 among objects disseminated within the environment with no centralized coordination. This paper presents an extended framework to enable u-KBs in mobile scenarios where a semantic discovery is carried out using metadata stored in RFIDs without fixed repositories. A dissemination protocol allows an on-demand retrieval of suitable descriptions directly from tags located on the objects. Such a vision allows to build a truly pervasive environment where autonomous objects compose a self-organized evolving discovery architecture suitable for u-Commerce purposes.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous; H.3.4 [Information Storage And Retrieval]: Systems and Software—Distributed systems, Information networks.

General Terms

Algorithms, Management, Languages.

Keywords

RFID, 802.11, U-Commerce, Pervasive Computing, Semantic Web, Resource Discovery, SOA, Knowledge Representation Systems

1. 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 in order 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 will interact with many computational devices simultaneously, extracting data from the objects "integrated" 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. This is achieved by exploiting a relatively large number of heterogeneous micro devices –e.g RFID (Radio Frequency IDentification) tags or wireless sensors– each conveying a small amount of useful information. Due to space, power and cost constraints, such devices are still currently endowed with low storage, little or no processing capability and short-range, low-throughput wireless links, though increasing efforts are being spent to increase such capabilities. Each mobile host in the 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. Pervasive computing calls for a decentralized and collaborative coordination between autonomous mobile hosts. As a consequence, pervasive knowledge-based systems have to achieve high degrees of autonomic capability from the knowledge management point of view, providing transparent access to knowledge sources that may be present in a given area. In [10], we devised solutions for the integration of semantic-enhanced EPCglobal RFID into Mobile Ad-hoc NETworks (MANETs). Semantic annotations were put into RFIDs attached to objects so that tagged goods stored a semantically rich description of the product the tag is clung to. This enabled objects equipped with RFID tags to describe themselves toward the "rest of the world" in a self-contained fashion. Nevertheless, we were still forced to use a fixed central component for reasoning over a Knowledge Base (KB). 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.

Here we show how 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. Resource discovery is carried out using metadata stored in RFIDs lacking fixed knowledge bases and inference tasks will be distributed among readers which provide minimal computational capabilities. Considering RFID technology, nowadays tags with higher memory availability disclose the concrete possibility of building cooperative environments where autonomous objects can be discovered, queried and inventoried without any support infrastructure. A proper dissemination protocol allows to exactly locate suitable descriptions directly on tags attached to objects thanks to a capillary diffusion of tag/reader basic information within the environment. Such a framework allows to build an environment where tagged objects and readers make a self-organized, autonomous discovery architecture. We consider a ubiquitous Knowledge Base (u-KB) as a distributed knowledge base whose individuals (assertional knowledge) are disseminated on the objects within the environment without centralized coordination.

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 RFIDs dipped in a MANET-based environment. RFID readers act as cluster-heads with respect to tags in their radio range (i.e. coordinate all communications) and they are able to automatically build up multi-hop inter-reader wireless communication when placed in the same area [11]. IEEE 802.11, 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 DLs (Description Logics) and originally conceived for the Semantic Web effort, particularly DIG [3], which is a more compact equivalent of OWL-DL (http://www.w3.org/TR/owl-features/). These design choices provide flexibility in architecture adaptation to specific application requirements. Moreover, reuse of Semantic Web standards simplifies the integration of u-KBs in larger information infrastructures. The feasibility of the approach has been proved by a simulation campaign performed using ns-2 (http://www.isi.edu/nsnam/ns/) network simulator.

The remaining of the paper is organized as follows: in the next Section we present motivation of our paper and application scenarios together with some significant aspects of the proposed approach. Section 3 outlines the data propagation and retrieval framework, whereas the ensuing case study further explains and motivates our proposal. Simulation results are presented and commented and conclusions close the paper.

2. CONTEXT AND APPLICATION SCENARIO

In current applications, 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 queries 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 mobile and ubiquitous computing applications. They are characterized by: user and device mobility; dependency on context; resource limitations. Hence, knowledge-based systems designed for wired networks are hardly adaptable, 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 requires shareable vocabularies, otherwise the lack of agreement about the meaning of descriptions would impair interactions among involved actors [12].

Few proposals for semantic-based annotation of physical commercial products can be found in the literature, beyond our previously mentioned work [10]. A solution applied to u-commerce environments was introduced in [17]. RFID tags, however, stored only a product code, which was used as a key to retrieve the corresponding semantic annotation from a fixed backend information system. This "virtual counterpart" approach [8], inherited from traditional RFID applications in supply chain management, does not allow true mobility and pervasiveness.

Figure 1 reports on the conceptualization of knowledge dissemination and discovery in our pervasive context. The u-KB layer provides common access to information embedded into semantic-enhanced EPCglobal RFID tags populating a smart environment. Hence, discovery and reasoning 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 Services) and translating remote methods into operations on the u-KB.

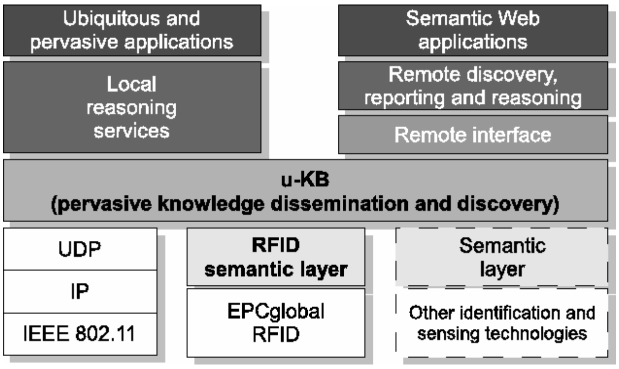
Ubiquitous and pervasive applications
Semantic Web applications
Local reasoning services
Remote discovery, reporting and reasoning
Remote interface
u-KB
(pervasive knowledge dissemination and discovery)
UDP
IP
IEEE 802.11
RFID semantic layer
EPCglobal RFID
Semantic layer
Other identification and sensing technologies

Figure 1: Architecture of the proposed framework

Advantages of semantic-aware approaches in RFID applications are widely acknowledged [4,14]. Traditional services the technology offers can be sensibly improved by introducing a semantically rich object description and discovery capabilities, without depending on centralized infrastructures. Let us consider the lifecycle of commercial products: manufacturing and quality control can 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 can be automated. Furthermore, sale depots can benefit from easier inventory management and can provide ubiquitous commerce (u-commerce) 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 [4]. 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.

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.

2.1 From KBs to u-KBs

In the following we assume the reader is familiar with basics of KRS and DLs [1]. In our approach we preserve the classical distinction between TBox and ABox.

The *TBox* (which provides the *conceptual* knowledge of the domain, e.g. apparel products) is expressed by means of an ontology document, which can be managed by one or more mobile hosts. Though it can be reasonably assumed that ontologies defined before object annotation and u-KB deployment seldom change during normal system activity, the TBox management can be performed also following paradigms and approaches usually devised in peer-to-peer protocols such as BitTorrent (http://www.bittorrent.org/beps/ bep_0003.html). 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

Ontology Universally Unique Identifiers (OUUIDs) [6] to unambiguously mark ontologies and to associate each individual to its reference ontology.

The *ABox* (providing the *factual* knowledge of a particular problem, e.g. describing the articles in an apparel store) 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, while objects of different categories may refer to different ontologies. In detail, each individual is characterized by:

- a globally unique item identifier (the EPC – Electronic Product Code – in the case of RFID tags);
- the OUUID;
- a set of data-oriented attributes, which allow to integrate and extend logic-based reasoning services with application-specific and context-aware information processing;
- semantic annotation, stored as a compressed document fragment in the DL-based language DIG.

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

Tell/Un-Tell operations are transparent 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 instant and by the ontology they refer to. The framework allows an autonomic and adaptive knowledge base maintenance, by means of a data alignment protocol between caches of different mobile hosts. Each node advertises individuals it sees in its proximity (e.g. via RFID). Other hosts store advertisements in their cache and forward them to hosts at one-hop distance. The protocol keeps track of the freshness of advertised individuals through sequence numbers, so that updates will be automatically propagated. Each individual has also a limited time-to-live, so that it will be automatically removed (un-told) from the u-KB if not renewed. Similar mechanisms are employed to disseminate ontology chunks within the network, aiming at a trade-off between fault tolerance and associated overhead. The system tends toward a steady state in which every host is aware of all ontologies and individuals in the environment. To reduce storage requirements and network load, semantic annotations of individuals are not included in advertisements; when needed they are provided on-demand.

Ask operations require a preliminary resource discovery step. The requester specifies the ontology identifier and a range for each particular attribute 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. On-demand provisioning of KB individuals minimizes transmissions for data alignment and prevents propagation issues in case of description update or individual removal. Additionally, filtering avoids unnecessary data transfers. Then the requester can fetch ontology chunks and filtered individuals by their respective providers, so that it can reconstruct a local subset of the whole KB, containing only the TBox and individuals which are actually needed. Finally, it can submit any Ask-type request to a local or remote reasoning engine.

3. BUILDING THE u-KB

Generally speaking, nodes participating to an ad-hoc network are moving and their inclusion within the MANET does not require any configuration procedure. Furthermore each node can 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. The communication between the tag field and readers exploits the semantic-enhanced EPCglobal RFID protocol data exchange [4], whereas the data propagation among readers is performed following a dissemination paradigm in IEEE 802.11 that we propose here. If two nodes are in radio range, they can directly interact, 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 shall not be depending on some other one to advertise/register object descriptions. Resources are autonomously exposed via the enhanced RFID and, in the same way, readers shall be able to discover them thanks to the preliminary propagation of data each cluster-head has seen in its range. Thus, the resource discovery process is based on three stages:

1. the extraction of good's parameters from RFID tag (to bring 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). This phase is performed after an on-demand request in unicast addressed to a selected node.

Each reader plays a central role in the whole Service Oriented Architecture (SOA) as it advertises contextual parameters referred to tags in its radio range (at the field layer) and 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 which will 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. Using that information, a mobile node directly selects a resource description referred to the field of a specified RFID reader.

Care has to be paid in the use of broadcasting mechanism to advertise object features. Its uncontrolled exploitation could be inefficient in terms of bandwidth usage and power consumption). In the proposed approach, only resource parameters are advertised in broadcast throughout the network in order to unambiguously identify both 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, advertisement flooding (due to the broadcasting mechanism) is reduced without sacrificing the correct use of a resource. As said, 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, only information strictly required for the unambiguous identification of a resource is shared within the network, whereas the semantically annotated description of the resource itself will be sent in unicast only to nodes explicitly interested in it.

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 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. Particularly, a requester attempting to rebuild a TBox will broadcast a specific 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. There, 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 currently reserved for future uses 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 reconstruction will reply in unicast with the managed chunk. Only when all the chunks become available the TBox will be ready for further exploitation.

Summarizing, the discovery procedure occurs in two steps. The first one is syntactic-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 descriptions of resources directly from the provider 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 can perform a substantial reduction of the induced traffic.

3.1 Protocol details

To perform the data dissemination, resource providers periodically send advertisement packets also specifying the maximum number of hops for the advertisement travel (MAX_ADV_DIAMETER). The cyclic advertisement diffusion allows to cope with tag and reader mobility. 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 discovery, a node attempts to cover the request by using resource descriptions stored within its own cache memory. If some semantically annotated description is missing, it can be retrieved in unicast using demand PDUs (Protocol Data Units) 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 the demand and solicit PDUs are used to update the cache memory of forwarding nodes. Figures. 2 and 3 show the typical sequence and involved actors of the data dissemination phase and of semantic annotations retrieval, respectively. After receiving required information from hosts in its search range, the requester newly performs the matchmaking. If the result is still unsatisfactory, the node can 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. In the following subsections, the discovery layer will be described in greater details. Structure of advertisement, request and solicit PDUs as well as the cache content organization of RFID readers will be explained.

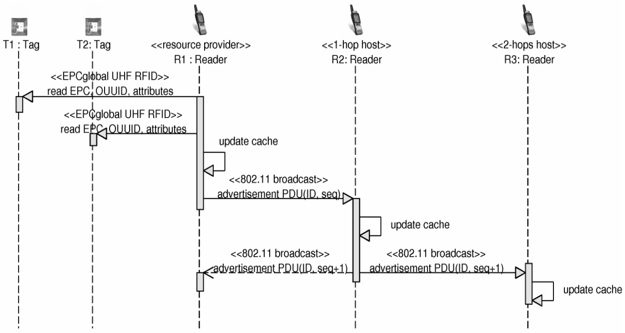

Figure 2: Tag data dissemination phase

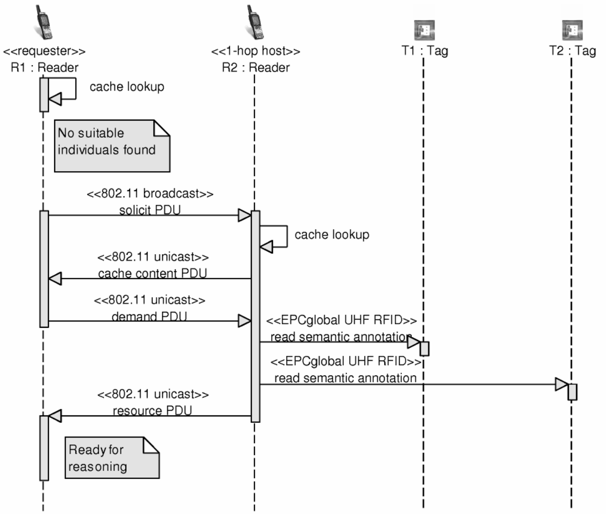

Figure 3: Retrieval of semantic annotations from tags

Advertisement PDU - In Figure 4 the structure of an advertisement PDU is sketched. All the resources inventoried by a reader are advertised by means of a unique advertisement PDU and then the size of the packet increases proportionally with the number of tags in the reader range. In what follows we outline PDU fields:

- TYPE: the kind of PDU (see Table 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.
- 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, for each tag it contains the OUUID value, the remaining life time of the resource, the maximum hops number for the advertisement travel, and finally the EPC code.

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

Table 1: PDU types exploited in the proposed framework

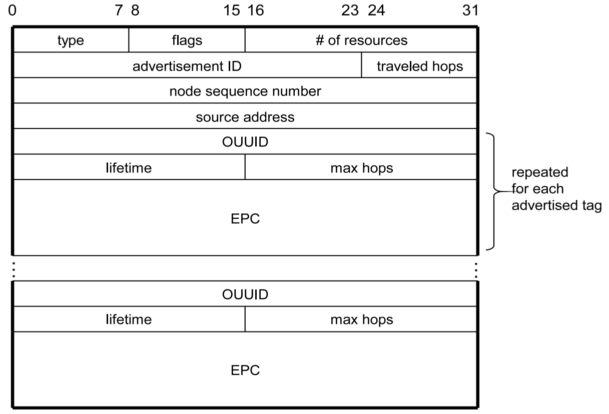

Figure 4: Advertisement PDU structure

A reader, which has inventoried a tag group, broadcasts an advertisement every DEFAULT_RTIME ms. 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 ms (less than DEFAULT_RTIME), it will retransmit the advertisement, assuming that a collision 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 the 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. This happens also when multiple readers attempt to scan the same tag. In particular, the entry is updated if the ADVERTISEMENT_ID is smaller than the one found in the packet. If the values are equal, the entry is updated if the stored TRAVELED_HOPS field is greater or equal to the just received one. 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. This means that the arrived PDU is more recent or has ran across a shorter path. 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 simple 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 before transmitting. After this check, the nodes verifies if the maximum advertisement diameter has been reached and, in case, builds the packet to be forwarded to its neighbors. It first writes into the new PDU its own sequence number into the NODE_SEQUENCE_NUMBER field. Then, for each service advertised 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). So a frame will be forwarded only if the increased value is lower than the one contained in MAX_HOPS field.

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 RFID readers and tags (1000 tags per reader). Since typical read rates are not above 100 tags/s for EPCglobal Gen-2 UHF RFID [5], we reasonably deem that the advertisement period DEFAULT_R_TIME should not be set below 30 s; moreover, let us set MAX_HOPS=4. Table 2 summarizes formulas and results of the analytical model. Value of input parameter a derives from Tab. 5, while c will be justified in the below 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 connectivity 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 the previous explanation of the data dissemination protocol, 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.

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)	$S_{adv} = (t/n)a+f$	20016
L_{min}	**Min traffic (B/s)**	$L_{min}=(n\,H / T_R)\,S_{adv}$	80064
L_{max}	**Max traffic (B/s)**	$L_{max}=(n^2 / T_R)\,S_{adv}$	600480

Table 2: Analytical model of the data dissemination protocol and performance bounds in a typical scenario

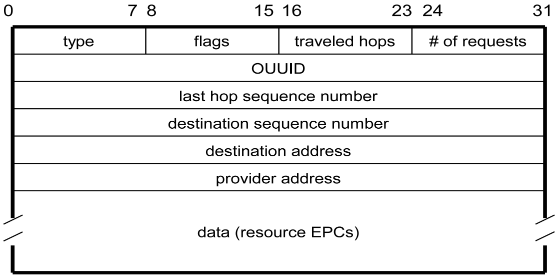

Figure 5: Demand PDU structure

Demand PDU - When starting a matching process, a node must first look within its cache table for entries compatible with the request and, in case, it must require in unicast the corresponding semantically annotated descriptions from their respective owners. A demand PDU as the one sketched in Figure 5 is used. In what follows the meaning of introduced PDU fields is summarized.

- TYPE: it is set to 3 (see Table.1).
- FLAGS: analogous to the corresponding field of the advertisement PDU.
- TRAVELED HOPS: how many hops the frame has already gone across.
- NUMBER OF REQUESTS: 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.
- DESTINATION ADDRESS: address of the last node processing the request.
- 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 resource descriptions at the same time and then waits for replies up to *ONE_HOP_WAIT*max_distance* ms (where *max_distance* indicates the maximum hop number between requester node and each provider). When this time has expired or all requested PDUs are received, the requester starts the proper matching stage.

Solicit PDU - If the requester has no resource descriptions in its cache or if managed resources are considered insufficient by the requester, 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 and we skip its structure. 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: it can exactly know the number of neighbors. So, before starting the matching or requesting new tag descriptions, it will wait for all the expected cache content PDUs for a t_w, defined as:

t_w = *ONE_HOP_WAIT* + *HOP_TRAVERSAL_TIME* * (*current_hops* - 1),

where *current_hops* is the hops number the solicit has to traverse up to reach the DISCOVERY_DIAMETER. Each node located at DISCOVERY_DIAMETER hops from the requester, after receiving a solicit PDU, 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.

Cache table management - Each reader manages a cache table where it stores information about both tags in its radio range and tags it has “seen” in the network. Figure 6 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 for the specific ontology.
- Lifetime: remaining time to live of a resource/tag.
- Timestamp: it marks the last reference to the entry (read/write). That is, this field is updated when a new resource is stored within the cache or when an existing one is invoked.
- Traveled hops: distance (hops number) between provider and cache holder.
- Sequence number: it is referred to the last resource provider.
- EPC: Electronic Product Code of a specific resource/tag.
- Resource description: the semantically annotated description of a resource. It will have a variable length, but in some case there can be a pointer to the compressed file containing the DIG description.

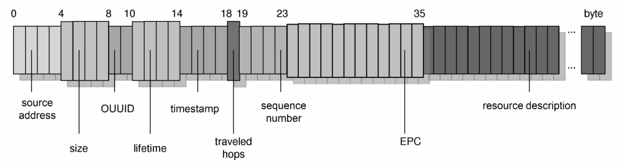

Figure 6: The structure of a record in the reader cache

An entry can be added to the cache table whenever the node receives an advertisement or a cache content frame arrives. Note that if a node receives an advertisement frame, the corresponding entry of the cache table is updated either if the sequence number within the packet is higher than the one stored or if the route the PDU suggests is shorter than the previously stored one. A cache content packet has a variable length according to the number of resource handles the PDU transports. Hence a cache update can involve more than one record. PDU fields are outlined hereafter.

- TYPE: it is set to 1 (see Table1).
- FLAGS: it maintains the ordinary meaning.
- N: the number of resources handles (and then cache tuples) the packet transports.
- REQUEST ID: identifier of the original request.
- OUUID: identifier of the reference ontology.
- LAST HOP SEQUENCE NUMBER: sequence number of the node sending the packet.
- DESTINATION ADDRESS: requester IP address.
- Resource records (one or more).

4. CASE STUDY: RFID-BASED u-KB

The proposed framework has been designed specifically for pervasive u-Commerce environments where a wide range of objects/products are endowed with RFID transponders conforming to the EPCglobal standard for class I - second generation UHF (Ultra High Frequency) tags [15]. Mobile RFID readers equipped with IEEE 802.11 wireless connectivity (from now on hosts) are responsible for u-KB creation and management. Tagged products represent KB individuals in our system. In a previous work [10] the EPCglobal standard was enhanced to support storage and retrieval on RFID tags of an ontology identifier, a set of attributes and a compressed semantic-based annotation, in addition to the EPC identifier. These basic elements recur in our definition of an individual in a u-KB, as explained above. Here the 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 [15]:

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 can be stored in additional TID memory from 20_h address. There, we store the OUUID identifier of the ontology referred by the semantic annotation in the tag. Finally, contextual attributes are stored within the User memory bank, together with the semantically annotated object description expressed in DIG. Due to verbosity of DIG language, the description is compressed using a specifically devised algorithm.

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. *Inventory* loop, where the reader detects one tag in range for each iteration and reads its EPC code;
3. *Access* to the memory content of a tag.

In our approach, original commands are exploited in novel ways, while keeping full backward compatibility. *Select* command is used to preselect semantic-enabled tags only. 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 tagged objects.

Let us consider the apparel department of a large shopping mall. Each article is tagged with a semantic-enhanced EPCglobal RFID transponder, storing: the EPC; the OUUID of the reference ontology for the apparel domain; data attributes such as price and size; a semantically annotated product description. For instance, a description such as "Italian striped black and white cotton shirt with red buttons, suitable for male young adults" could be fully expressed in an annotation referring to classes and properties defined in the apparel ontology (see [10] for details).

Each shelf integrates an RFID reader and is equipped with IEEE 802.11 connectivity. Shelf hosts form a MANET and collectively build the u-KB in the department, which allows semantic-based product search for nearby users equipped with a 802.11-enabled mobile device. In such scenario, interaction with EPCGlobal RFID technology takes place in various fashions:

1. Dissemination. After each advertisement period DEFAULT_RTIME, a shelf host scans RFID tags in its range to detect apparel products in the shelf. Only semantic-enabled tags are preselected, by means of a *Select* command with parameters as shown in Table 3. 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.

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

Table 3: SELECT command able to detect semantic-enabled tags

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

Table 4: READ command able to extract OUUID from the TID memory bank

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

Table 5: READ command to extract contextual attributes from User memory bank

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 (e.g. in case of shelf replenishment), it will read the OUUID and contextual attributes, exploiting two Read commands, as shown in Table 4 and Table 5, respectively. Data extracted from the RFID tag is 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 (e.g. in case of sold product), the host will wait for the TTL to expire before removing the entry. This prevents the well-known issue of “RFID event flickering” [8] 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.

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

Table 6: READ command to extract the compressed semantic annotation from User

2. Discovery. A customer searching for apparel items will issue a demand PDU to a shelf node in the radio range of his mobile device. His demand will contain the OUUID of the apparel ontology and acceptable ranges for data attributes, e.g. price between 20$ and 50$ and medium to large size. When a host receives the demand 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 semantically annotated compressed object description stored in the User memory bank of the tag, with a Read command as in Table 6. 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. After collecting the resources that satisfy the parameters of his demand, the user can start a semantic matchmaking process in order to find the best ones w.r.t. desired product characteristics [10]. Semantic user request is expressed according to the ontology similarly to product descriptions.

3. Ontology provisioning. As already proposed in [10], the EPCglobal Object Naming Service (ONS) [16] can be used as a fallback mechanism for ontology support in ubiquitous computing contexts if an Internet connection is available. In our u-KB system, if a requester does not manage an ontology X (or part of it), if it is endowed with an Internet connection, it can query the ONS with URI_X derived from $OUUID_X$ and fetch the corresponding compressed ontology file or a chunk of it.

4. Implications of RFID data locality. In typical pervasive computing scenarios, the content of a particular RFID tag can be relevant 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 specific application requirements. In the above shopping mall example, a different u-KB will be built in each store department. 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 backend warehouse information system.

4.1 Simulations and Results

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.
3. duration of service discovery sessions.

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 [2] has been adopted.

Host motion follows the random waypoint model [7] 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 RFID 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 time. 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 conditions of single scenarios (e.g., high link breakage ratio or network partitions).

Figures 7 and 8 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 contents and request packets are produced on-demand by client nodes and their neighbors. Furthermore, Figure 7 shows that the number of generated packets decreases with the increase of pause time. As nodes remain stationary for less time, probability increases that radio links are lost and advertisement PDUs are not echoed: hence advertisers schedule PDUs for retransmission, so increasing the overall traffic.

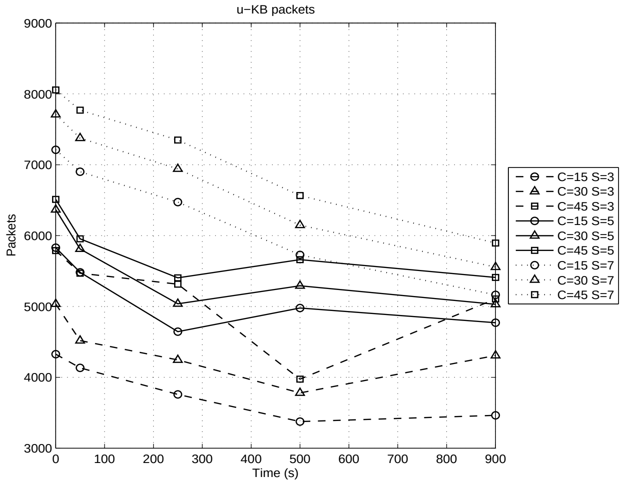

Figure 7: Dissemination and discovery u-KB packets, dependence on pause time

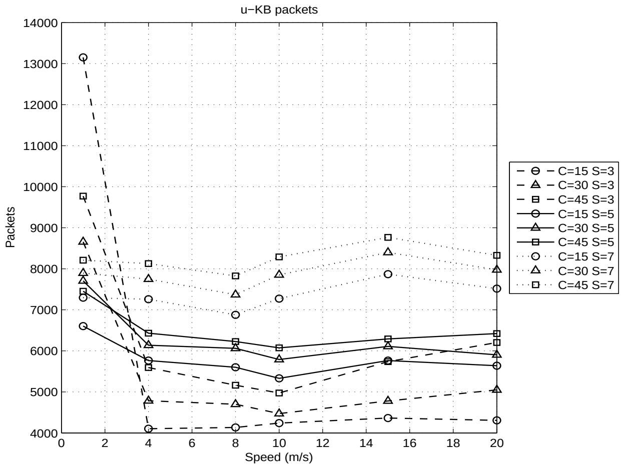

Figure 8: Dissemination and discovery of u-KB packets, dependence on maximum speed

The hit ratio is reported in Figure 9 and Figure 10, 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, these results 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, with more providers the capillarity of data dissemination is increased. Second, the proposed resource discovery mechanism induces by itself a load balancing of requests within the network, since the closest hosts with available resources will be contacted first by a requester.

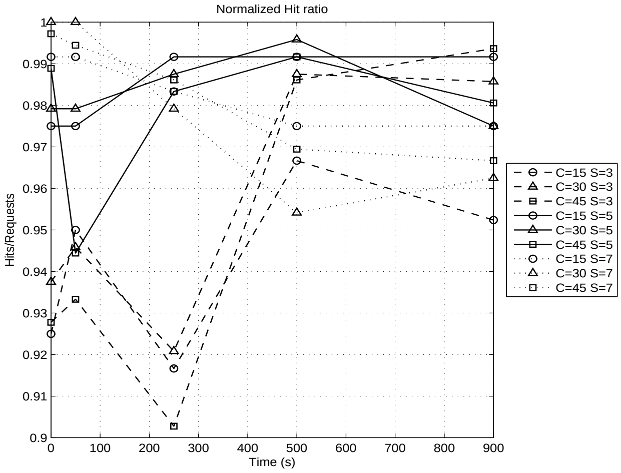

Figure 9: Hit ratio, dependence on pause time

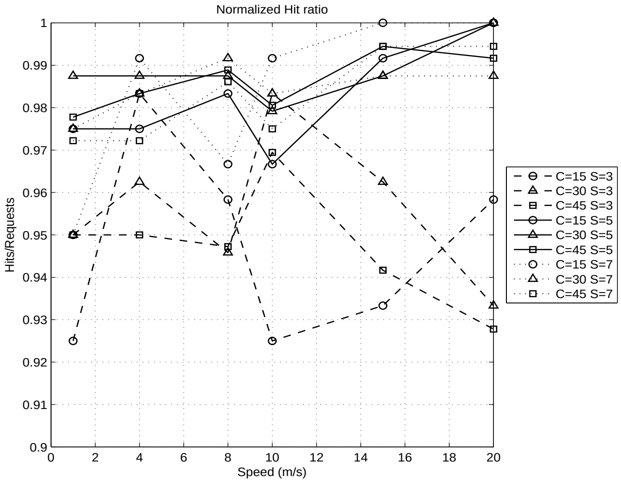

Figure 10: Hit ratio, dependence on maximum speed

Figure 11 and Figure 12 report the time values for a successful resource discovery session. Analysis shows that, for a given number of servers, the service time decreases as the number of clients increase. This is due to another feature of the proposed protocol, namely that, when a solicit PDU is answered by cache content PDUs, intermediate nodes cache the resource records. This reduces latency in the response to later requests. Overall values are still slightly high with respect to the requirements of pervasive computing scenarios. This result may be influenced by the fact that the current protocol implementation is not particularly optimized for execution speed. Further techniques should be devised, however, in order to reduce duration of resource discovery sessions.

5. CONCLUSIONS

We presented an extended framework to enable u-KBs in mobile scenarios where a semantic discovery is carried out using metadata stored in RFIDs without fixed repositories. A dissemination protocol allows an on-demand retrieval of suitable

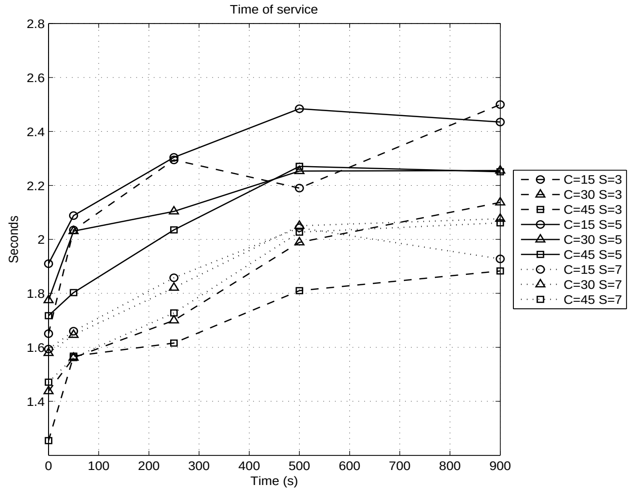

Figure 11: Time for obtaining a hit, dependence on pause time

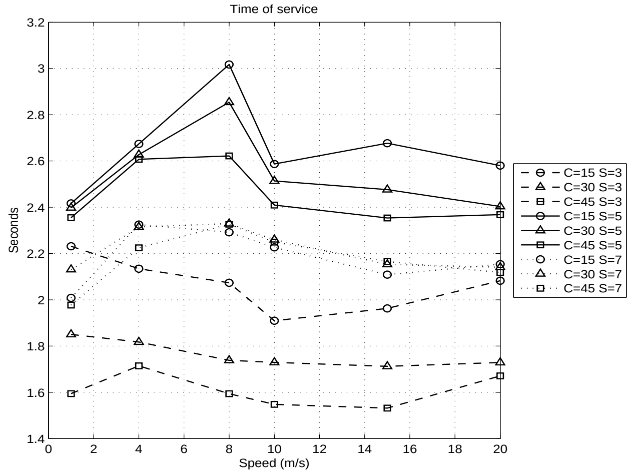

Figure 12: Time for obtaining a hit, dependence on maximum speed

descriptions directly from tags located on the objects. An advanced resource discovery framework is supported by a knowledge dissemination protocol, so allowing an “on-demand” retrieval of suitable descriptions directly from tags located on the objects. Such a framework allows to build a truly pervasive environment where autonomous objects compose a self-organized evolving discovery architecture suitable for u-Commerce purposes An analytical model has been developed for a preliminary assessment of resource requirements. A thorough evaluation of the approach using ns-2 simulation environment has been proposed in order to test the feasibility of the proposed approach.

6. REFERENCES

[1] Baader, F., Calvanese, D., Mc Guinness, D., Nardi, D., Patel-Schneider, P. *The Description Logic Handbook*. Cambridge University Press. 2002.

[2] Bakre, A., Badrunath, B.R. I-TCP: indirect TCP for mobile hosts, *Proceedings of the 15th International Conference on Distributed Computer Systems*, pages 136-143, Vancouver, BC, 1995.

[3] Bechhofer, S., Möller, R., Crowther, P.. *The DIG Description Logic Interface*. Proc. of DL’03. 2003

[4] Di Noia, T., Di Sciascio, E., Donini, F.M., Ruta, M., Scioscia, F., Tinelli, E.. Semantic-based Bluetooth-RFID interaction for advanced resource discovery in pervasive contexts *International Journal on Semantic Web and Information Systems*, 4(1): 50-74, 2008.

[5] Kawakita, Y., Mistugi, J. Anti-collision performance of Gen2 air protocol in random error communication link. *Proceedings of the International Symposium on Applications and the Internet Workshops - SAINT 2006*, pages 68–71, 2006.

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

[7] Resta G., Santi P. . An analysis of the node spatial distribution of the random waypoint model for ad hoc networks. In *Proc. of ACM Workshop on Principles of Mobile Computing (POMC)*, pages 44–50, 2002.

[8] Römer, K., Schoch, T., Mattern, F., Dübendorfer, T. Smart Identification Frameworks for Ubiquitous Computing Applications. *Wireless Networks*, 10: 689–700, 2004.

[9] Ruta, M., Di Noia, T., Di Sciascio, E., and Donini, F. Semantic-Enhanced Bluetooth Discovery Protocol for M-Commerce Applications. *International Journal of Web and Grid Services*, 2(4):424–452, 2006.

[10] Ruta, M., Di Noia, T., Di Sciascio, E., Piscitelli, G., and Scioscia, F.. RFID meets Bluetooth in a semantic based u–commerce environment. In *9th International conference on Electronic Commerce, ICEC 07*, pages 107–116. 2007.

[11] Ramanathan, R., Redi, J.. A brief overview of ad hoc networks: Challenges and directions. *IEEE Communications Magazine* 40, 20–22. 2002.

[12] Vasudevan V. Ensembleware: contextual service provisioning in an ubiquitous services world. *International Symposium on Applications and the Internet 2004*, pages 10, 2004.

[13] Watson, R., Pitt, L., Berthon, P., Zinkhan, G. . U-Commerce: Expanding the Universe of Marketing. *Journal of the Academy of Marketing Science*, 30: 333–347, 2002.

[14] Weinstein, R. RFID: A technical overview and its application to the enterprise. *IT Professional*, 7: 27–33, 2005.

[15] Traub, K., Allgair, G., Barthel, H., Bustein, L., Garrett, J., Hogan, et al. EPCglobal Architecture Framework. *EPCglobal Ratified specification*. 2005.

[16] EPCglobal Inc. Object Naming Service (ONS} - ver. 1.0 . *EPCglobal Ratified specification*. 2005.

[17] Maass, W., and Filler, A. Towards an infrastructure for semantically annotated physical products. *Informatik*, 94:544—549, 2006.