HAL
open science

Content Discovery in Heterogeneous Mobile Networks

Diego Borsetti, Claudio Casetti, Carla-Fabiana Chiasserini, Luigi Liquori

To cite this version:

Diego Borsetti, Claudio Casetti, Carla-Fabiana Chiasserini, Luigi Liquori. Content Discovery in Heterogeneous Mobile Networks. E. Hossain. Heterogeneous Wireless Access Networks: Architectures and Protocols, Springer, pp.419-441, 2009, 10.1007/978-0-387-09777-0_15 . hal-00909620

HAL Id: hal-00909620

https://inria.hal.science/hal-00909620

Submitted on 26 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1

Content Discovery in Heterogeneous Mobile Networks*

Diego Borsetti[1], Claudio Casetti[1], Carla-Fabiana Chiasserini[1], and Luigi Liquori[2]

[1] Dipartimento di Elettronica, Politecnico di Torino, Italy
Email: lastname@tlc.polito.it
[2] INRIA Sophia Antipolis Méditerranée, France
Email: Luigi.Liquori@inria.fr

1.1 Introduction

Today, it is not uncommon for the inhabitants of major cities to have a large selection of network technologies, both wired and wireless, at their fingertips. Hence, to have an unprecedented number of services readily accessible.

Whereas the availability of wired technologies is limited to one at a time (e.g., an office LAN, a residential xDSL), as far as wireless technologies are concerned, people are usually exposed to several of them simultaneously (e.g., GSM/GPRS and UMTS cellular networks, WiFi, Bluetooth). Although users commonly rely on one technology at a time for a specific service, considerable research effort has been devoted to inter-operability among technologies, with the aim of granting more bandwidth to users and, potentially, lower fees.

Among recently-developed mobile applications (e.g., the Google Mobile suite, the Traffic Message Channel or the Intermodal Journey Planner), a widespread interest is surrounding those that enable the collection and sharing of various information contents for users on the move. The growth of such services will foster the creation of "virtual communities" of users sharing similar interests and goals: as way of example, students on a campus who share class material, tourists browsing through local attractions listings, car drivers seeking and exchanging traffic information, or pedestrians accessing their domestic or working environment via hand-held devices.

Taking into account the previous remarks, it is clear that a fundamental problem in wireless networks is the discovery and sharing of information among users, which can relay on more than one communication technology. While in the context of wired networks, many algorithms and protocols have been presented for content discovery, much fewer proposals target heterogeneous wireless environments.

In this chapter, we first review some of the most relevant solutions presented for content discovery, in both wired and wireless contexts (Sections 1.2 and 1.3). We focus on the well-known *publish/subscribe* asynchronous messaging paradigm and

* This work is supported by AEOLUS FP6-IST-FET Proactive, and by the *Regione Piemonte* through the project VICSUM.

discuss the benefits of designing an overlay network relying on an underlay wireless network. We then introduce a possible network architecture, composed on the one hand of infrastructured nodes, such as WLAN access points and cellular system base stations, and on the other hand of mobile nodes that are equipped with multi-interface wireless terminals, namely, pedestrian users as well as vehicles (Section 1.4). We devise how an overlay network can be created and efficiently implemented on such a communication network using the publish/subscribe paradigm and its content-based routing algorithms. We describe the semantics and the interaction among the logical network entities as well as the possible interface selection strategies (Section 1.5). Finally, we take a heterogeneous wireless networks, including the IEEE 802.11 technology and the UMTS cellular network as a case study, and we show some performance results of the publish/subscribe approach in such a scenario (Sections 1.6 and 1.7).

1.2 Content discovery in dynamic networks

The explosive growth of the information technology fosters the design of communication networks that let users discover and share information, anywhere and anytime. In a wireless, urban environment, information sought by users can be acquired either from server nodes belonging to the backbone infrastructure or from other users, by exploiting cooperative mechanisms [1, 2]. Such a communication system, however, poses several technical challenges, which have been recently addressed both within research projects and in the scientific literature.

Examples of projects focusing on the support of innovative services within an urban environment are IntelCities [3] and WikiCity [4]. IntelCities is a European project that develops efficient and innovative e-government services for citizens. WikiCity, instead, is aimed at distributing and processing real-time data coming from electronic devices scattered throughout the city and connected to the Internet infrastructure.

In the literature, solutions proposed for wired peer-to-peer (P2P) networks, such as Gnutella, Freenet, Fastrack, or eDonkey, do not apply to mobile scenarios because of the dynamic nature of the network where users continuously move and contents may appear and disappear. A promising approach is instead the publish/subscribe messaging paradigm [5], an asynchronous, many-to-many communication model designed to distribute information to a large number of users adopting either a content-based or a topic-based routing. According to the publish-subscribe messaging paradigm, users can be both information providers (publishers) and information consumers (subscribers). Consumers subscribe to the publish/subscribe system and specify the type of information that they are interested in, while producers publish data to the system. The system disseminates the information to all (if possible) the consumers that are interested in receiving it, according to the interests they declared. Having decoupled users in publishers and subscribers allows for a great flexibility and scalability in presence of networks with a dynamic network topology (i.e., with high node churning). The selection of messages for reception and processing is called

filtering. There are two kinds of filtering, namely topic-based or content-based, each associated to a specific routing. In a nutshell, topic-based routing broadcasts all published messages (on some topic) to all users subscribed to that topic; content-based routing delivers to subscribers only messages that exactly match subscriber-defined attributes or contents. Many publish/subscribe systems have recently been developed for wired networks (e.g., Xnet [6, 7], Siena [8] or Gryphon [9]), as well as for wireless networks [10, 11, 12, 13, 14, 15, 16, 17, 18]. In particular, the work in [11] specifies the way consumers and producers are matched together, i.e., by applying a cross-layer approach that leverages some routing-specific metrics, such as hop count or node traffic load. In [12], the cooperative downloading strategy, named SPAWN, addresses peer discovery and content selection. Peer discovery uses a centralized approach, as in BitTorrent [19], and Slurpie [20], as well as a distributed one that leverages the broadcast nature of the wireless medium and allows nodes to overhear information about the content availability at neighbors. Content selection is determined by a proximity-driven piece selection strategy, where proximity estimation is based on hop count. The work in [14] combines the publish/subscribe approach with a content-based routing scheme. In [16], instead, the authors present a dynamic publish/subscribe system for mobile peer-to-peer environments, which integrates an extended on-demand multicast routing protocol and content-based messaging.

A valuable solution to effectively implement the publish/subscribe approach is to define an overlay network, which is built on top of the physical one, such that two neighbor nodes in the overlay may be many links apart in the physical network. In general, overlay networks are capable of providing a rich spectrum of services through the use of aggregated computational power, storage and contents [21, 22, 23]. The main idea is therefore to achieve information discovery and retrieval via a seamless, geographically distributed, open-ended network of bounded services owned by the overlay components.

In the field of wireless networks, the publish/subscribe paradigm and overlay networks are exploited together in [24, 25, 15]. In [24, 25], service discovery protocols are based on the deployment of a virtual backbone of directories within an infrastructure-less network. Each node composing the backbone performs service discovery in its proximity, while global service discovery is provided by the cooperative action of the directories. In [15], an overlay network is conceived to operate in a mobile ad hoc network: the overlay network that routes events from publishers to subscribers dynamically adapts itself to the changing topology by means of cross-layer interactions.

In the context of wired computer networks, an interesting solution, named Arigatoni [26], has been designed to provide a large variety of services through a multi-layer overlay network. In a nutshell, in a multi-layer overlay network, the responsibility assigned to network nodes differs. Super-peers, called Brokers, act as servers for a subset of peers (named colony). Ordinary peers, called Agents, submit queries to their Broker and receive results from it. Brokers are also connected to each other; they route messages across the overlay network, submit, delegate, and answer queries on behalf of the Agents in their colony. This structure is replicated *recursively*, creating an *n-layer topology*. Arigatoni provides service discovery with variable guar-

antees in a virtual organization, where peers can dynamically appear, disappear, and self-organize. Furthermore, it is a fully–programmable overlay network: it dictates how and where services are declared, discovered and orchestrated (via a simple business language) in the overlay, allowing peers to ask, provide and use global services and resources.

Thanks to the above features, Arigatoni appears to be a suitable choice for information delivery and sharing in a mobile environment. We therefore define a urban mobile network scenario, whose nodes are equipped with multiple wireless interfaces and, inspired by Arigatoni, we devise an overlay network for information discovery and retrieval in such an environment. For the sake of completeness, some further details on the Arigatoni solution are provided below.

1.3 Arigatoni in a nutshell

What follows is a short overview of the activity of the main entities and of the protocols involved in Arigatoni (the interested reader can refer to [27, 28, 26]).

1.3.1 Functional units

Two main logical entities (the Agent and the Broker) and two basic protocols (a registration and a service discovery protocol) are the core of the Arigatoni overlay network.

The Agent is a computing device with wired/wireless connectivity capabilities. It does not necessarily need to be a high-end device, such as a supercomputer; on the contrary, it may have limited storage and computation capabilities, and few basic installed applications (a simple editor, one or two compilers, an email client, a mini browser). Agents are organized in Colonies, led by a Broker. Unlike the Agent, though, the Broker is required to be a mid- to high-end device, equipped with a high–speed wired/wireless connection and a service table, crucial to perform the publish/subscribe content-based routing. Given the hierarchical overlay, colonies may recursively be embedded into super-colonies, each led by a super-Broker.

The Agent

It should be able to work in *local mode* for all the tasks that it can manage locally or in *colony mode*, by first registering itself to one or many colonies of the overlay, and then by asking and serving colony-originated requests via the Brokers. The tasks of an Agent can thus be summarized as:

- discovering the address of one or more Brokers, acting as colony leaders, upon its arrival in a “connected area”;
- registering to one or many Brokers, thus entering the virtual organization;
- requesting and offering services to other Agents, through its own Broker;
- connecting directly to other Agents in a peer-to-peer fashion, and exchanging services between each others. Note that an Agent can also be a service provider. This symmetry is one of the key features of Arigatoni.

The Broker

It requires higher capabilities than an Agent to store and manage the content-based routing table of the colony it leads. Such table is essential to route queries, and the Broker must also efficiently match and filter the routing table against a received query. The tasks of a Broker are:

- discovering the address of another Broker, and possibly embedding its colony into the other Broker's;
- registering/unregistering Agents in its colony and updating the internal content-based routing table accordingly (who offers what within the colony, its address, and other geographical information);
- receiving Agents service requests, discovering the services that satisfy an Agent request in its local colony, according to its content-based routing table, or delegating the request to its direct super-Broker;
- in case the Agent request can be satisfied, forwarding, in a service response, all the information necessary to allow the requesting agent to communicate directly with the agent offering the service;
- in case the agent request cannot be satisfied, notifying the requesting agent, after a fixed timeout period, that its service request could not be served.

There are mostly two mechanisms of service discovery, namely:

- the process of a Broker finding and negotiating services to serve an Agent request in its own colony;
- the process of an Agent discovering a Broker, upon physical/logical insertion in a colony.

The first discovery is processed by Arigatoni's service discovery protocol, while the second is processed out of the Arigatoni overlay, using well-known network protocols like DHCP, SLP in Bluetooth or Active/Passive Scanning in WiFi.

The Service Discovery Protocol (RDP)

It is used by a Broker to find and negotiate services to serve agent requests in its own colony. RDP allows the request for multiple services and service conjunctions (i.e., each agent may offer several services at the same time). The RDP protocol allows Agents to:

- ask to a Broker a request for a service set S;
- reply to a Broker the availability to offer a service set S'.

The colony's Broker handles the service request received through RDP and looks up the service set in its routing table, filtering S against the offered set S'. If a match is found, the Broker returns to the requesting agent the address of the agent matching, partly or fully, the request. From then on, the two agents interact in a peer-to-peer fashion, without further intervention of the Broker using a simple coordination language (e.g., the BPEL language [29]).

Each Broker maintains a content-based *routing table* locating the *services* that are registered in its colony. The table carries one entry for each member matching the ID of the Agent with the set of services it can offer.

The Virtual Intermittent Protocol (VIP)

It manages peers' participation in Arigatoni's colonies. The protocol deals with the *dynamic topology* of the overlay, by allowing individuals to login/logout to/from a colony. Registration is the act through which an agent becomes member of a colony. An Agent is allowed to unregister when it has no pending service requests or offers. Agents that abruptly unregister or behave as "free riders" (using other Agents' services without offering or giving theirs in return) are tagged as unfair and may be forcefully unregistered from a colony by its Broker.

An Agent registers to a colony with a list of services. If a Broker accepts an individual in its colony, then it sends a service update to its direct super-Broker in order to communicate the availability of the new services in its colony. This message is then propagated from Broker to Broker until the root (if any) of the multi-layer overlay is reached. This means a high node churn forces routing tables to be *faulty* until all service updates are properly propagated. As such, service registration in an overlay network computer is an activity that must be taken seriously into account.

1.4 A mobile, heterogeneous network scenario

Let us consider an urban area in which a mobile network is deployed by using an ad hoc communication technology (e.g., WiFi). A further coverage of the network area is obtained through a cellular network, such as GSM/GPRS or UMTS. Such mobile network is populated by both mobile users, e.g., pedestrians with hand-held devices, cars equipped with browsing/computational capabilities, public-transportation vehicles and roadside infrastructures such as bus stops. All devices have multiple wireless interfaces. Depending on their mobility, they may also be equipped with a wired interface. Such is the case of wireless Points of Access (PoA), which are installed at bus stops, in order to provide connectivity either to users waiting for a bus or to the bus itself (hence to its passengers). PoAs are thus equipped with a wired and one or more wireless interfaces. In such setting, devices carried by cars and pedestrians play the role of Mobile Agents; PoAs, such as roadside infrastructures and public transportation vehicles (buses, trams, cabs,...), act as Brokers. In the urban mobile and dynamic scenario described above, it is possible to define an overlay network based on the publish/subscribe messaging paradigm and content-based routing. We describe this overlay network, named Arimove, in the next section.

1.5 Content discovery in heterogeneous wireless network: the Arimove system

Arimove, firstly introduced in [30], provides an efficient mapping between physical devices in the wireless underlay network and virtual entities in the overlay network.

Below, we present an overview of the overlay network followed by a description of the main entities and their interaction.

1.5.1 Overlay overview

A Broker is implemented at a fixed infrastructure in the urban environment. A suitable choice could be a bus stop, since metropolitan transportation companies are likely to bundle bus stops with electric power and, in some cases, even with fixed network connectivity. A Broker colony is composed of Mobile Agents that have registered to it when they were within radio range of the PoA installed at the bus stop. Therefore, a colony is a logical entity, whose members may be physically located anywhere within the area where Arimove is deployed.

However, to take into account the high mobility of the scenario and enhance its performance in terms of load balancing and service response time, we introduce an additional, Arimove-specific entity, the *Mobile Broker* (mB). This unit may be a public transport vehicle equipped with a scaled-down Broker-like wireless device. Every Mobile Broker is *associated* to (i.e., it has the same identity of) a single Broker. Clearly, at the underlay level, connectivity between the Mobile Broker and the associated Broker may at times be severed.

The main aim of the Mobile Broker is to introduce the novel concept of *colony–room*: a small subset of Mobile Agents with a wireless connection to the Mobile Broker (e.g., pedestrian/vehicles around a bus/cab or traveling along the same direction of the bus/cab during a traffic jam...). In addition, thanks to its mobility, the Mobile Broker can collect registrations from Mobile Agents that were too far from the PoA of the associated Broker, and, therefore, might had never had the chance to register to it.

The Mobile Broker collects (un)registrations, service requests and service offers from the Agents within the colony–room. When a wireless connection has been established between the Mobile Broker and a roadside PoA (not necessarily corresponding to the associated Broker), the data path to the associated Broker is again available and an information exchange takes place resulting in the updating of each other's data. Specifically, the following actions occur. Firstly, the associated Broker merges the Mobile Broker's routing table with the one it currently carries. Then, the associated Broker handles the registration/discovery information and generates the appropriate responses. Finally, depending on the response time, the responses are returned to the Mobile Broker before it leaves the wireless PoA coverage, or the next time it connects to a PoA.

Figure 1.1 illustrates the relationships among overlay and underlay entities. A central coordination entity is located at a headquarter (HQ), in our case corresponding to the local transportation authority building. The coordination entity plays the role of a super-Broker and it is provided with a wired connection to each of the 4 roadside PoA at bus stops (B_1 to B_4). Mobile Brokers (mB_1 and mB_3) shuttle between bus stops, each carrying a different Broker association (to B_1 and B_3), while Mobile Agents (portable devices in the figure) are either connected to Brokers or

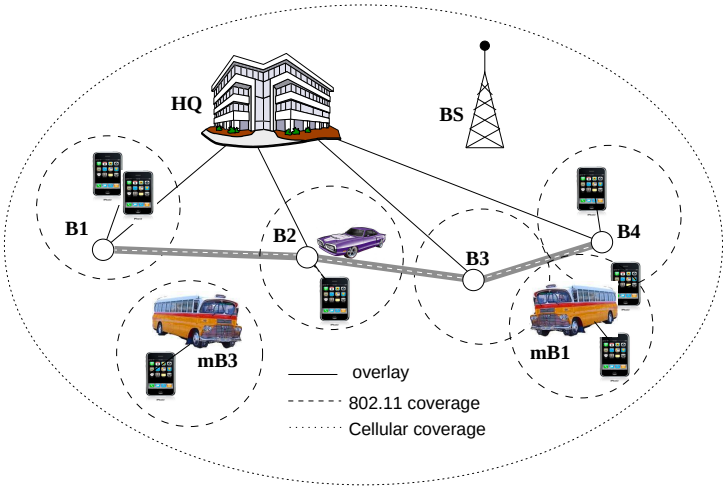

Fig. 1.1. An example of heterogeneous network scenario

Mobile Brokers, depending on their mobility. A base station (BS), belonging to a cellular network system, provides a wireless coverage on the whole area.

1.5.2 Overlay entities

We can now summarize the different activities of the three main entities in Arimove, i.e., Mobile Agents, Brokers and Mobile Brokers.

Mobile Agents

Their activity is carried out through the following main operations.

- *Broker Discovery*: if the underlay is a broadcast network, such as a 802.11 WLAN, HELLO messages are issued by one or more Brokers (and/or Mobile Brokers)[3]; their reception allows an Agent to perform the choice of the Broker to which to register using information stored in the HELLO message itself. Broker discovery in a non-broadcast network may be performed through the use of a directory service offered by a well-known host readily available on the network.
- *(Un)Registration*: the Agent sends a service registration (SREG) message of the VIP protocol trying to register to a new Broker or to unregister from the current Broker.
- *Service Request/Response*: these tasks require that the Agent is already registered to a Broker and that it is part of a colony. It can send a service request (SREQ) of the RDP protocol to its Broker or a service response (SRESP) offering some services. The service will be then exchanged in peer-to-peer fashion.

[3] namely, HELLO messages from a broker may be sent on the broadcast channel of a cellular network, but it entails the direct cooperation of the cellular carrier/operator

Brokers

Their activity is carried out through the following main operations.

- *Colony Health*: in a broadcast underlay, periodically broadcasting of HELLO messages to let Mobile Agents discover them;
- *Colony (Un)Registration*: to and from a higher-level Broker;
- *Colony Management*: the Broker interacts with colony members, (un)registering and handling service requests through the VIP and RDP protocols.

Mobile Brokers

The activity of a Mobile Broker is carried out through the following three main operations.

- *Broker Association*: the process of associating to a specific Broker; the Broker for which the Mobile Broker acts as colony-room is chosen according to some policy (see below) and the association is held throughout the Mobile Broker's route.
- *Colony-Room Advertising*: periodical broadcasting of HELLO messages along its whole route; HELLO messages forward information about the associated Broker for which they are acting as colony-room.
- *Relaying*: relaying VIP and RDP messages from (to) Agents inside the colony-room to (from) the associated Broker. Relaying may occur at once if a wireless connection to a PoA exists, or it may be deferred until the wireless connection is re-established.

1.5.3 Membership policies

As a byproduct of the Arimove overlay, members of a colony will be geographically distributed, although this distribution should be carefully planned by a Broker (accepting or refusing registration requests) for load balancing purposes.

VIP registration policies are usually not specified in the protocol itself; thus every Broker is free to choose its acceptance policy. Different self-organization policies may be used to address issues such as load-balancing and colony specialization. Possible policies therefore are: (i) *mono-thematic*: a Broker accepts an Agent in its colony only if it offers services that the colony already owns in large quantities, so as to increase its specialization; (ii) *balanced*: a Broker accepts an Agent in its colony only if it offers services that the colony lacks, with the aim of evening out its service offer; (iii) *unbalanced*: a Broker unconditionally accepts all Agent registrations. Membership to a colony is also affected by the Mobile Broker association. The choice of which Broker is associated to a Mobile Broker can be performed by a specific load balancing algorithm that is run periodically, i.e., when the public transportation vehicle leaves the deposit and sets off on its route. One possible aim of the load balancing algorithm is to let the Mobile Broker collect Agents for a Broker with a scarcely populated colony at the time of the Mobile Broker departure; other aims, such as specializing the colony population, can be also envisaged.

1.5.4 Overlay service discovery

Service discovery over a MANET, hence in Arimove, plays a crucial role in the successful retrieval of information. There are two main concerns regarding the issue of service discovery. The first one is to *expedite* the selection of the Mobile Agent providing the service, given that Agents are nominally free to roam in and out of their own Broker's underlay reach. Therefore, if a service is advertised by more than one Agent, and a request for that service is pending, a Broker should be given the opportunity to hand the service request over to the Agent that is more likely to be within its reach. The second concern is the *suitability* of the match that the Broker is about to create. Indeed, finding a "good" Agent carrying the requested service must also account for its ability to establish an effective communication channel with the Agent requesting the service. The mobility of both Agents (the requesting one and the potential provider) must be accounted for, e.g., by selecting Agents that are either within radio range of each other, or that are likely to remain within some PoA coverage for enough time. In publish/subscribe jargon, the "filtering" matching function between potential subscriber and publishers must take also into account the "reachability" at the underlay network level.

From a practical standpoint, service discovery at an Arigatoni Broker is carried out through a table that mainly records colony member IDs and their service lists. In Arimove, however, the table information for each Agent is integrated by a *liveliness* field, indicating the time elapsed since the last contact from that Agent, and by a *mobility* field, that can be used to pinpoint the position of the Agent and to infer the direction of its movement (the latter information is provided by the Agent in its last message sent to its Broker).

In order for these additional table parameters to be maintained up-to-date, Agents are expected to interact with their own Broker on a regular basis. Such interaction, in the form of a refresh SREG, should not be limited to the period when the Agent is within its own Broker radio range. Rather, it should also be promoted when the Agent is within *any* Broker range (whence it will be relayed to the Agent's Broker). The refresh SREG will therefore be issued by the Agent upon hearing a HELLO message from a Broker[4]. The rationale is to let the Agent's Broker know that the Agent is within coverage of whatever wireless technology is used by any Brokers, hence it is readily accessible if a service is requested. It should be noted that an Agent may choose to refrain from sending refresh SREG if the underlay is charging a per-access fee. For each Broker, the content-based routing table has the form:

Agent ID	Services	Liveliness	Mobility
user	$\{S_i\}^{i=1\ldots n}$	t	(x, y, θ, v)
$\ldots$	$\ldots$	$\ldots$	$\ldots$

where $\{S_i\}^{i=1\ldots n}$ is the set of services it can offer, t is the time passed since the Agent has sent a message, and (x, y, θ, v) is a quadruple denoting physical position,

[4] Broadcast storms are prevented by forcing a latency period between consecutive SREG from the same Agent.

Table 1.1. Comparison of the three underlay networks under study in terms of various performance metrics

Underlay	Overlay connection	Capacity	Energy	QoS	Pricing
IP	AP to HQ	high	low	high	low
UMTS	Mobile Agent to AP	med	med	med	high
WiFi	Mobile Agent to mB and AP	low	high	low	low

direction and speed (all those information easily retrievable by a GPS module, or computed through it). The table is updated according to the dynamic registration and unregistration of Agents in the overlay. When an Agent asks for a service, then the query is *filtered* against the routing tables of its own Broker; in case of a *filter-failure*, the Broker forwards the query to its direct super-Broker.

1.6 A case study: IEEE 802.11 and UMTS coexisting technologies

The underlay network we envision hinges on different PoAs. As already mentioned, a bus stop is a suitable place for a WiFi hot spot, but it clearly is not a viable choice to encase a UMTS base station (commonly termed a *Node B*). Both types of PoAs are however expected to route incoming traffic to an IP-based fixed network that guarantees connectivity among the different elements of the architecture. Also, the two wireless technologies have different features in terms of QoS, energy consumption and, mostly, pricing. Table 1.1 compares the three underlay networks in terms of logical connectivity (from-to), bandwidth capacity, energy consumption, quality of service, and pricing.

Therefore, a strategy should be devised in order to select the "appropriate" wireless underlay network. In many cases, one may need to switch from WiFi to UMTS and vice-versa because of signal loss, radio hardware problems, pricing issues, or required quality of service (QoS). Moving to one underlay network wireless to another is usually transparent to the Arimove overlay network. Indeed, in either case, RDP and VIP traffic is routed toward the overlay peers. Switching may be automatic when both wireless networks are available, or may be suggested or imposed by the Broker or by the Mobile Agent. Moreover, in the registration phase, the Mobile Agent and the Broker may agree on which underlay to use, whether it is WiFi, UMTS, or a smooth combination of both.

At the time of the first VIP registration, the Mobile Agent communicates its physical position and speed (x, y, θ, v) to the Broker. The Broker and the user negotiate the choice of the underlay (e.g., WiFi connection in a downtown zone until the available bandwidth is not dangerously low). The rationale of the strategy is simple and obeys to two (natural) rules:

1. When a Mobile Agent moves to one coverage area to another, then either it continues to stay connected with the Broker (in the abandoned area) using UMTS,

or it is invited to change colony, i.e., the Broker sends to it a SREG logout unless it has some pending requests to serve or receive.

2. When a Mobile Agent moves from one WiFi zone to another *in the same area*, then it essentially has two ways to stay connected to the Arimove virtual organization: either using ad hoc MANET protocols at the cost of poorer performance, or switching to the UMTS network until another WiFi zone is detected. The choice of the underlay protocol is in principle left to the mobile user unless the Broker forces the choice of one network instead of another.

This simple strategy induces the following considerations: moving from one area to another "forces" the Mobile Agent to change broker, while moving in the same area "advises" the Mobile Agent to stay in the colony it is connected to. The overlay therefore tries to limit as much as possible high node churning that, as is well-known, contributes to slow down overlay performance.

1.7 Performance evaluation

Below, we show the performance of the Arimove architecture obtained under the case study network introduced in Section 1.6. We first describe the simulation scenario and then present the results.

1.7.1 Simulation scenario

We implemented Arimove in the Omnet++ [31] simulator, coding the overlay part and exploiting the existing wireless underlay network modules. In the underlay, we used IEEE 802.11 at the MAC layer and the DYMO routing protocol (an AODV-like reactive routing protocol). The UMTS RACH (Random Access Channel) and dedicated channel were also simulated.

We tested the performance of Arimove in a mobile environment. We considered 3 Mobile Brokers and 120 Mobile Agents. We assumed that the nodes are randomly scattered in a 1-km-wide square city section and move around according to the random walk mobility model. Every node is assigned a speed of v m/s, randomly selected in a specified interval (see below).

Upon entering the topology, a node acting as Mobile Agent owns a set of S unitary services (e.g., files, traffic information, point of interests) randomly chosen from a set of C services. A service is included in a node subset with probability p, equal for all services and all nodes, so that each node owns an average of $\mathrm{S} = p{\cdot}C$ services. In our simulations, we choose $C = 20$ while p is a varying parameter.

A Mobile Agent issues a (SREQ) for a service it is missing. The inter-request time of each missing service is supposed to be exponentially distributed with parameter λ. It entails that the larger the number of services a node is lacking, the more frequently it issues an SREQ message. To further clarify, using the above notation, p is inversely proportional to the average SREQ rate.

If an SREQ is successful (i.e., the Broker returns to the requesting Agent the address of another Agent who is willing to provide the requested service), the two

peers establish a connection and a download is started. We will assume that a peer engaged in a download (either as sender or receiver) rejects any SREQ or any further connection request *coming from the same type of underlay* until the download is complete. In our simulations, the size of a downloaded file is set to 500 Kbytes.

The simulated city topology features six 802.11 PoAs, each corresponding to a Broker. A single UMTS cell covers the entire simulated area. The wireless channel is considered as always reliable, and no obstacles (such as high-rise buildings) are assumed to be present. Brokers apply the unbalanced acceptance policy and filter the routing table against a received SREQby using the liveliness information only.

1.7.2 Simulation results

Although the number of parameters that can potentially be analyzed is quite high, we focus on few selected sets of results that highlight the performance of the system in presence of either 802.11 coverage or UMTS coverage, as well as in a scenario featuring both technologies in the same area.

We initially address the success probability of a service request in an 802.11-only coverage scenario, i.e., the probability that a service request is positively answered by a Mobile Agent either in the requester's colony or in another colony (after delegation to the super-Broker). Figure 1.2 carries a comparison of success probabilities for an Arimove system as a function of p. The top plot shows results for two different inter-request times (the node speed is set between 0 and 1 m/s), while the bottom plot allows us to examine different node speeds (with a 600 s inter-request time). As expected, higher content availability entails more successful retrievals, regardless of request rate or node speed. As expected, a longer inter-request time, hence less congested underlay network, results in a higher retrieval success probability. Also, it is quite meaningful to underscore that higher speeds negatively affect the retrieval due to nodes leaving the 802.11 coverage more frequently and thus being unable to complete a request/response cycle that they may have initiated.

We point out that some underlay-related events are responsible for missing responses: for example, a miscue from the routing protocol may lead an Agent to wrongly believing that it is within radio range of a Broker (or that a Broker may be reachable in multi-hop fashion through other nodes); an SREQ will therefore be issued but never delivered to the Broker, negatively affecting the success probability.

Another metric of interest is the average time after which a response to an SREQ is returned to the requesting Agent (Figure 1.3). Again, we first look at an 802.11 scenario. As can be expected, some of the trends already observed for the success probability are found in the response times as well. However, all curves peak around $p = 0.4$. As can be recalled, p and SREQ rates are inversely proportional, so values lower than $p = 0.4$ counterbalance the scarcity of resources with a high SREQ message rate that "keeps alive" many routing paths; this avoids the high overhead of the path setup process typical of reactive routing protocols (such as DYMO). In addition to it, a smaller success probability results in fewer peer-to-peer downloads overloading the network. On the contrary, values higher than $p = 0.4$ exhibit a lower discovery time thanks to the wealth of services available throughout the network.

When the attention shifts to a scenario featuring UMTS-only coverage, the underlay becomes less of an issue. The probability of successful retrieval is very close to 1, the only failures being due to connection requests towards Agents that have already become engaged in a download with another Agent. Similarly, the discovery time is very low, as shown in Figure 1.4, since the only significant latency comes from RACH access (and no multihop connection, along with its routing overhead, is involved).

Finally, we look at a scenario that combines a spotty 802.11 coverage and an "umbrella" UMTS coverage. Clearly, if the UMTS operator charges a per-access fee, while the 802.11 access is free or available at reasonably hourly fees, Agents will strive to connect to the latter. Therefore, in areas where both coverages are available, Agents will try and connect through the 802.11 underlay. If an Agent cannot connect to an 802.11 PoA (either directly or through a multihop path), it will use UMTS to access the Arimove system. In our simulations, we considered one of the setups used in previous tests, (i.e., Agent speed between 1 and 2 m/s, $p = 0.6$) and analyzed three cases, each with a different 802.11 coverage of the overall area (either 25% or 75%).

Interestingly, the probability of resorting to the UMTS network is directly mapped onto the coverage percentage for higher loads (i.e., inter-request times equal to 100 ms), as can be seen in Figure 1.5. Therefore, if, on average, 75% of the Agents are outside the 802.11 PoA coverage (802.11 coverage = 25% in the figure), the probability of switching onto UMTS matches such figure. However, as the load decreases, the number of Agents who successfully complete an SREQ cycle on 802.11 becomes larger, since multihop connectivity even *outside* the radio range of a PoA is no longer hindered by high traffic on the underlay network.

Figure 1.6 presents the probability of successful retrieval observed for SREQ cycles that are performed on 802.11 only (lower curves, with 25% and 75% coverage), and the one observed for all SREQ cycles, i.e., including those that are routed over the UMTS network (higher curves). In the 802.11-only message exchanges, beside the usual increase in success probability as the inter-request time increases, we also observe that smaller coverages yield a better success rate than larger coverages. This is due to fewer downloads on the underlay network (most occur over UMTS, therefore they do not hinder SREQs on 802.11). Clearly, the very high completion probability under UMTS positively affects the overall performance.

The last set of results, shown in Figure 1.7, attains to the average time to satisfy a service request, computed over all SREQ cycles. Depending on the coverage, the use of UMTS rather than 802.11 boosts the performance, lowering the discovery time for 802.11 coverage at 25% (when the majority of Agents is forced to use UMTS).

Conclusions

This chapter addressed the problem of content discovery in heterogeneous mobile networks. It focused on a network architecture, composed on the one hand of infrastructured nodes, such as WLAN access points and cellular system base stations, and

on the other hand of mobile nodes, that are equipped with multi-interface wireless terminals. The study adopts the well-known *publish/subscribe* paradigm and designs an overlay network, relying on the underlay wireless network, to implement such a messaging paradigm. The chapter described the semantics and the interaction among the logical network entities, as well as the possible interface selection strategies. Finally, the benefits of using multiple wireless technologies for content retrieval in a mobile environment were highlighted through some performance results. The results were obtained for a case study including WiFi hot spots and UMTS base stations as network infrastructure entities.

References

1. F.H.P. Fitzek, M. Katz, *Cooperation in wireless networks: Principles and applications – Real egoistic behavior is to cooperate!,* Springer, 2006.
2. M. Dohler, D.-E. Meddour, S.-M. Senouci, and A. Saadani. Cooperation in 4g: hype or ripe. *IEEE Technology and Society Mag.* in press.
3. The InterCity project. `http://intelcities.iti.gr/intelcities`.
4. The WikiCity project. `http://senseable.mit.edu/wikicity`.
5. P. Th. Eugster, P. Felber, R. Guerraoui, and A. M. Kermarrec. The Many Faces of Publish/Subscribe. *Computing Survey*, 35(2):114–131, 2003.
6. R. Chand and P. Felber. A scalable protocol for content-based routing in overlay networks. In *Proc. of NCA*, 2003.
7. R. Chand and P. Felber. XNet: A Reliable Content-Based Publish/Subscribe System. In *Proc. of SRDS: Symposium on Reliable Distributed Systems*, 2004.
8. A. Carzaniga, D.S. Rosenblum, and A.L. Wolf. Design and Evaluation of a Wide-Area Event Notification Service. *ACM TOCS*, 19(3), 2001.
9. G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R.E. Strom, and D.C. Sturman. An efficient multicast protocol for content-based publish-subscribe systems. In *Proc. of ICDCS*, 1999.
10. L. Iftode, C. Borcea, N. Ravi, and T. Nadeem. Exploring the Design and Implementation of Vehicular Networked Systems. Technical Report DCS-TR-585, Rutgers.
11. A. Varshavsky, B. Reid, and E. de Lara. A Cross-Layer Approach to Service Discovery and Selection in MANETs. In *IEEE MASS*, Washington, DC, 2005.
12. A. Nandan and *et al.* Co-operative downloading in vehicular ad-hoc wireless networks. In *WONS '05*, pages 32–41, St. Moritz, Switzerland, January 2005.
13. M. Fiore, C. Casetti, and C.-F. Chiasserini. Efficient retrieval of user contents in MANETs. In *IEEE INFOCOM '07*, May 2007.
14. D. Lundquist and A. Ouksel. An Efficient Demand-driven and Density-controlled Publish/Subscribe Protocol for Mobile Environments. In *ACM International Conference on Distributed Event-based Systems*, pages 26–37, Toronto, Canada, 2007.
15. M. Avvenuti, A. Vecchio, and G. Turi. A cross-layer approach for publish/subscribe in mobile ad hoc networks. In *MATA*, pages 203–214, 2005.
16. E. Yoneki and J. Bacon. Dynamic publish/subscribe in mobile peer-to-peer systems. In *OTM Workshops*, pages 1–2, 2004.
17. R. Meier and V. Cahill. Steam: Event-based middleware for wireless ad hoc network. In *ICDCS Workshops*, pages 639–644, 2002.

18. R. Meier, V. Cahill, A. Nedos, and S. Clarke. Proximity-based service discovery in mobile ad hoc networks. In *DAIS*, pages 115–129, 2005.
19. BitTorrent, Inc. `http://www.bittorrent.com/`.
20. R. Sherwood, R. Braud, and B. Bhattacharjee, Slurpie: A cooperative bulk data transfer protocol, In IEEE INFOCOM '04, Hong Kong, March 2004.
21. Globus Alliance. `http://www.globus.org/`.
22. JXTA Community. `http://www.jxta.org/`.
23. OSGi Alliance. Open Services Gateway Initiative. `http://www.osgi.org/`.
24. U.C. Kozat and L. Tassiulas. Network Layer Support for Service Discovery in Mobile Ad Hoc Networks. In *IEEE INFOCOM*, pages 1965–1975, San Francisco, CA, April 2003.
25. F. Sailhan and V. Issarny. Scalable Service Discovery for MANET. In *IEEE PerComm*, Kauai Island, Hawaii, March 2005.
26. L. Liquori and M. Cosnard. Logical Networks: Towards Foundations for Programmable Overlay Networks and Overlay Computing Systems. In *TGC, Trustworthy Global Computing*, Lecture Notes in Computer Science. Springer, 2008.
27. D. Benza, M. Cosnard, L. Liquori, and M. Vesin. Arigatoni: Overlaying Internet via Low Level Network Protocols. In *JVA, John Vincent Atanasoff International Symposium on Modern Computing*, pages 82–91. IEEE, 2006.
28. R. Chand, M. Cosnard, and L. Liquori. Powerful resource discovery for Arigatoni overlay network. *Future Generation Computer Systems*, 24(1):31–38, 2008.
29. D. Jordan and J. Evdemon. *Web Services Business Process Execution Language Version 2.0*. OASIS Web Services Business Process Execution Language (WSBPEL) TCO.
30. L. Liquori, D. Borsetti, C. Casetti, and C. Chiasserini. Overlay Networks for Vehicular Networks. Research report, INRIA Sophia-Antipolis Méditerranée, 2008.
31. OMNET++ Discrete Events Simulator. `http://www.omnetpp.org`.

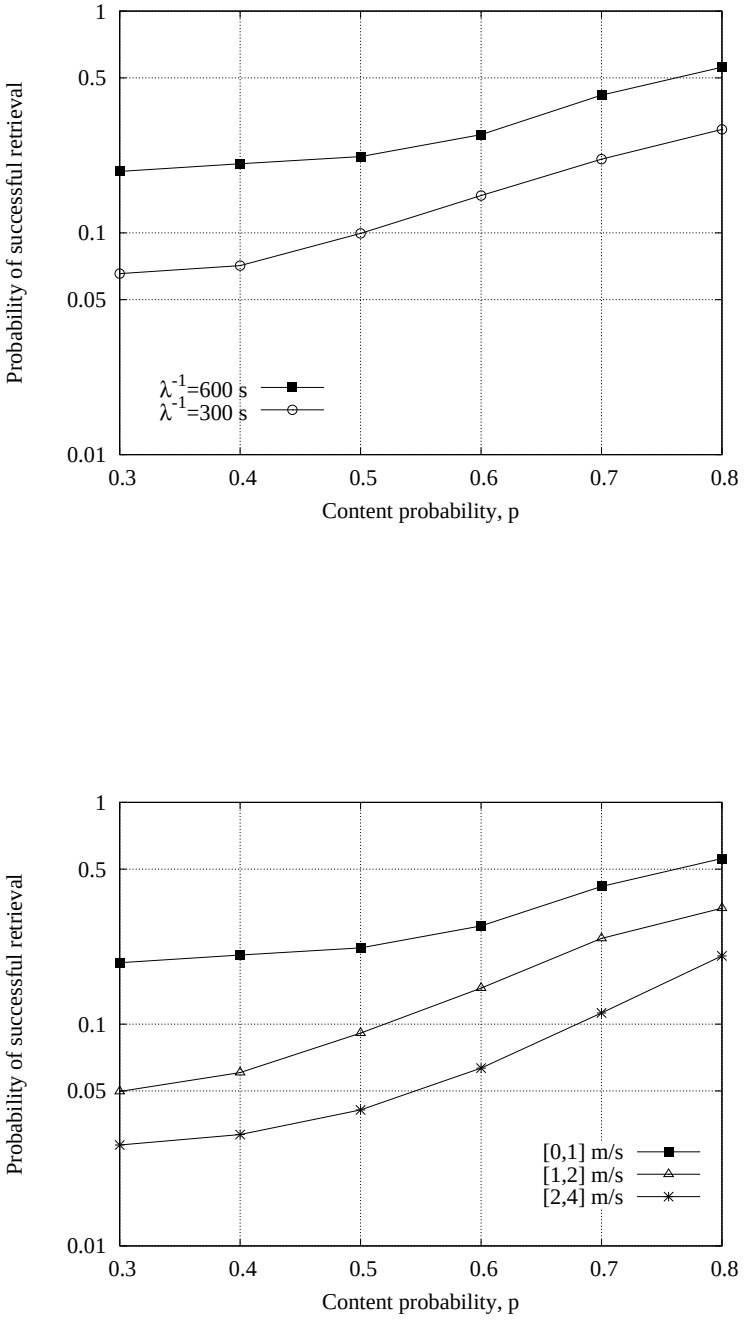

Fig. 1.2. Success probability of a service request as a function of the initial content availability at each user. Results are derived for different request rates (top) and different user speeds (bottom). 802.11 scenario

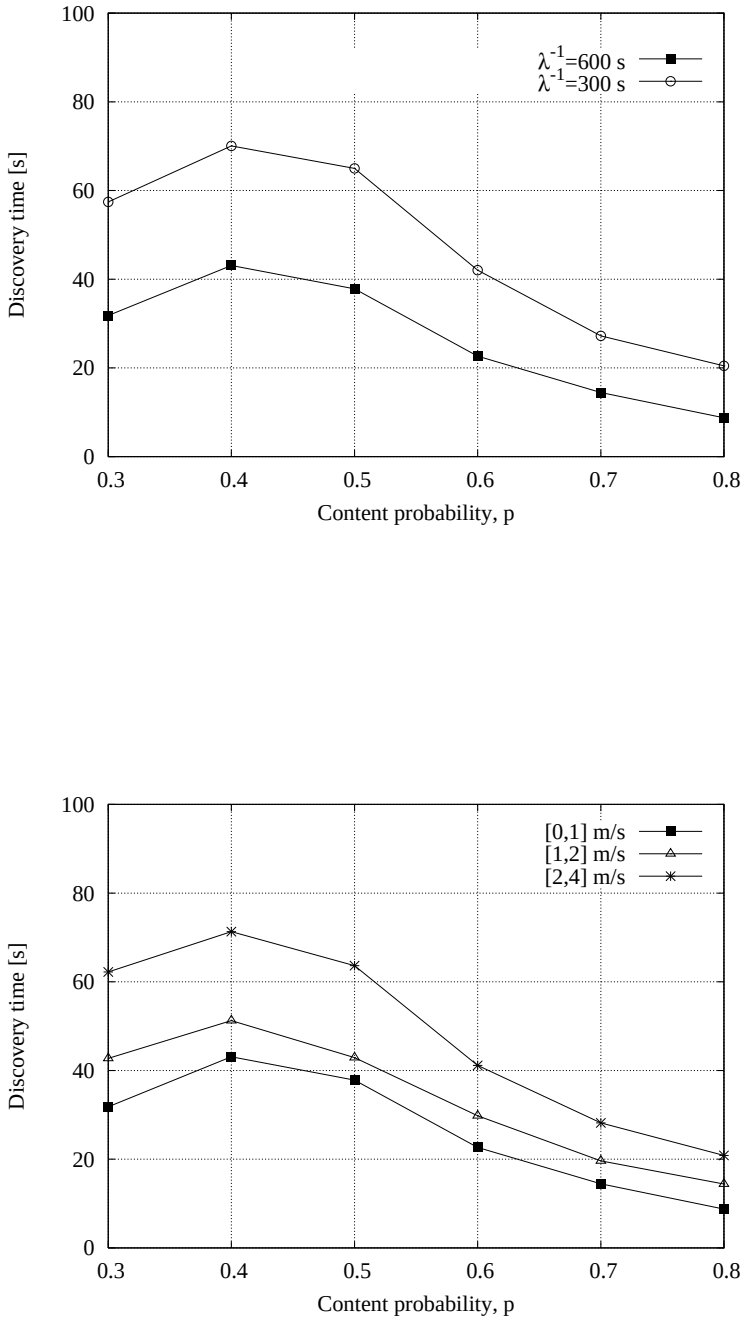

Fig. 1.3. Average time to satisfy a service request as a function of the initial content availability at each user. Results are derived for different request rates (top) and different user speeds (bottom). 802.11 scenario

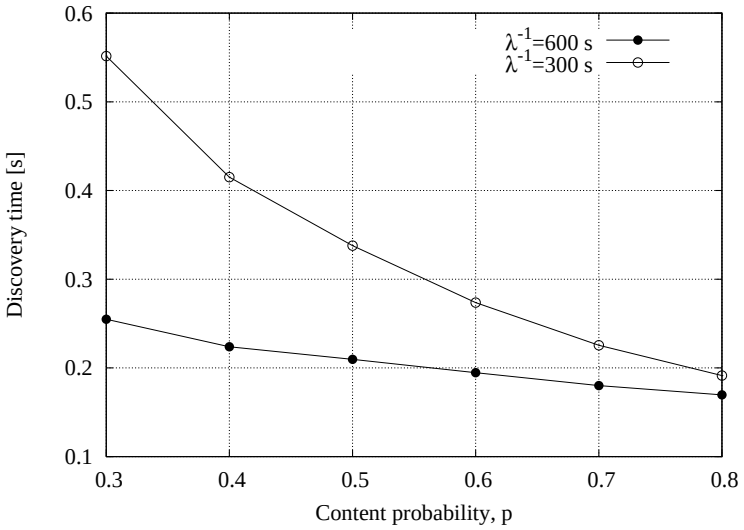

Fig. 1.4. Average time to satisfy a service request as a function of the initial content availability at each user. Results are derived for different request rates. UMTS scenario

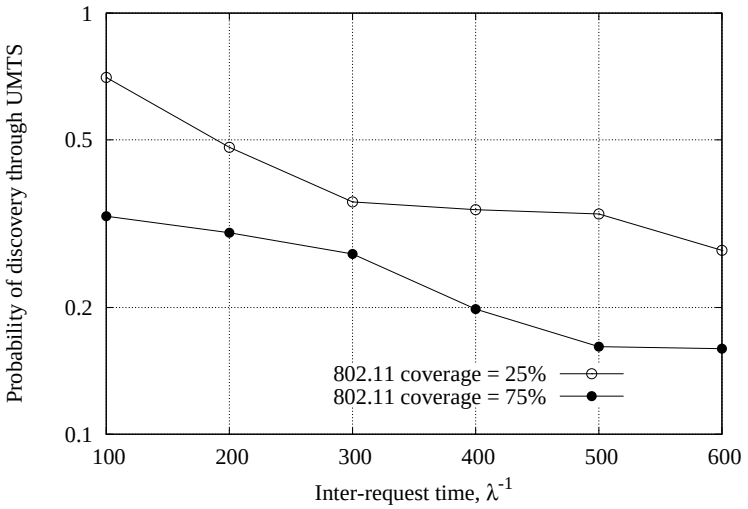

Fig. 1.5. Probability that a SREQ has to be routed over UMTS due to unavailability of an 802.11 route to the Broker, as a function of inter-request time. 802.11/UMTS mixed scenario and different 802.11 coverages.

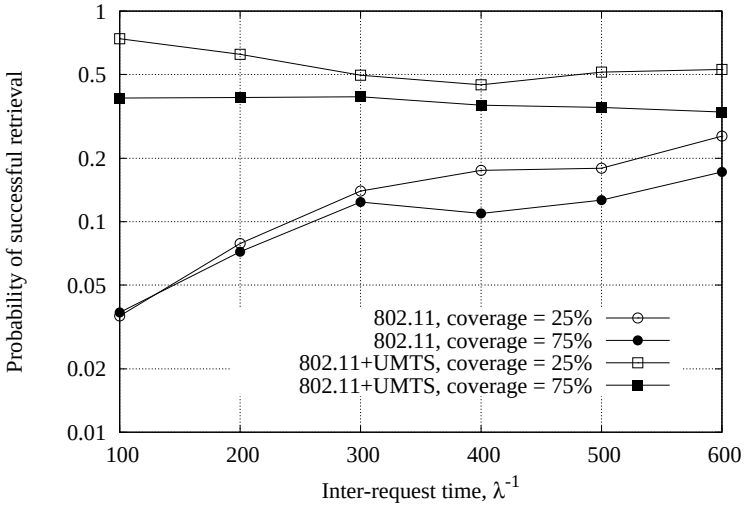

Fig. 1.6. Probability that a SREQ issued is successful, as a function of inter-request time in 802.11/UMTS mixed scenario and with different 802.11 coverages.

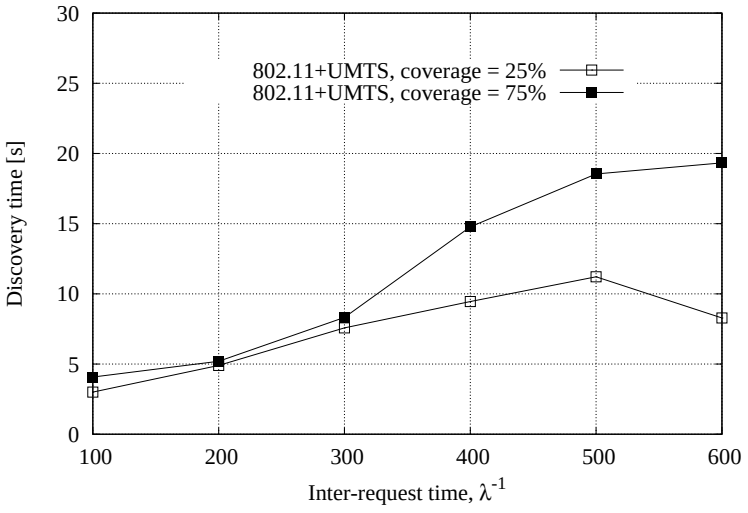

Fig. 1.7. Average time to satisfy a service request as a function of inter-request time. 802.11/UMTS mixed scenario and different 802.11 coverages.