Performance evaluation of a discovery and scheduling protocol for multihop ad hoc mobile grids

Antônio Tadeu Azevedo Gomes[1]*, Artur Ziviani[1], Luciana dos Santos Lima[2], Markus Endler[3]

[1]National Laboratory for Scientific Computing – LNCC,
Av. Getúlio Vargas, 333, 25651-075, Petrópolis, RJ, Brazil

[2]Nokia Siemens Networks (NSN)
Av. das Américas, 3434 - Bloco 7- 3º. andar, 22640-102, Rio de Janeiro, RJ, Brazil

[3]Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente, 225, 22453-900, Rio de Janeiro, RJ, Brazil

Abstract: Despite the many research efforts addressing the integration of mobile nodes into grids, only a few of them have considered the establishment of mobile grids over wireless ad hoc networks (hereafter, *mobile ad hoc grids*). Clearly, such grids need specialized resource discovery and scheduling mechanisms. To the best of our knowledge, though, the research on these mechanisms for mobile ad hoc grids is still preliminary. Besides, and more importantly, it has approached discovery and scheduling as separate mechanisms, which, we argue, is not suitable for mobile ad hoc grids. In this paper, we propose the integration of resource discovery and scheduling for mobile ad hoc grids into a single protocol called DICHOTOMY (DIscovery and sCHeduling prOTOcol for MobilitY). This protocol allows computational tasks to be distributed appropriately in a mobile ad hoc grid, while mitigating the overhead of discovery messages exchanged among the nodes. Our experiments show that the protocol: (i) does proper scheduling, allowing an efficient load balancing among the nodes and helping with lowering the average completion time of tasks; (ii) keeps the discovery efficiency at acceptable levels in mobility scenarios and (iii) scales very well with respect to an increasing number of nodes, both in the total amount of energy savings due to packet transmissions and the distribution of such savings among the nodes.

Keywords: mobile grids, resource management, self-organizing networks.

1. Introduction

There has been an increasing amount of research over the past few years on wireless grids[36]. These extend traditional grids[2] by integrating mobile devices in a wireless infrastructured network as either resource consumers or providers in the computing infrastructure. Only a few efforts[31,45], however, have addressed the more challenging issue of dynamically establishing spontaneous, *purely* mobile ad hoc grids i.e. grids established over mobile ad hoc networks, and even these have achieved only preliminary results. Further development in mobile ad hoc grids can be justified by some applications that demand high computational power but at the same time have to be used at places or conditions where network infrastructure may be or may suddenly become unavailable[21]. In particular, we focus on application scenarios in which the mobility of devices is constrained to pedestrian (walking) speeds. Examples of such scenarios include:

- *Emergency response systems* for disaster handling and crisis management situations. For instance, consider a large rescue and medical team working in a natural disaster scenario, such as the one caused by the Indian Ocean tsunami in Dec. 2004 or the hurricane Katrina that hit New Orleans in Aug. 2005. In such scenarios, the seamless integration of computational resources from on-site mobile nodes can be crucial for rapidly achieving advanced forms of collaborative work, as for example to collect and automatically process information about groups of injured people (e.g. for triage) and thus better allocate rescue teams and medical resources[13]. In this context, mobile ad hoc grids can be regarded as specializations of the more general concept of hastily formed networks[9].

- *Field research systems* dynamically deployed on isolated areas. For instance, consider a team of hydrogeologists throughout a large semi-arid region, such as the Northeastern Brazil. Such a team can exchange data about underground water resources (e.g. hydraulic head) being collected by their mobile nodes, and may use the computational resources of such nodes for *in-loco*, preliminary numerical analysis on the collected data, so as to simulate and predict some aquifer condition of interest (e.g. the effect of irrigation developments).

Both aforementioned application scenarios demand the formation of *multihop* mobile ad hoc networks (hereafter,

*e-mail: atagomes@lncc.br

MANETs). MANETs allow mobile nodes to self-organize into arbitrary and temporary topologies that expand the basic communication range of–and consequently the service offering to these nodes. In the case of mobile ad hoc grids, the basic service provided by the nodes is the capability to allocate computational resources to execute tasks. The expanded range offered by MANETs permits that more nodes share computational resources, but for this to be feasible specialized resource discovery and scheduling mechanisms are needed. In the context of this paper, resource discovery comprises gathering information about resources that mobile nodes make available in the MANET, whereas resource scheduling comprises selecting the most appropriate nodes (from the point of view of resource provisioning) to execute a particular set of tasks. We highlight that our focus in this paper is on the resource discovery and the *initial* scheduling of tasks to declared available resources. Hence, we do not consider an eventual unavailability of a resource *after* it has been discovered, selected, and is already running a submitted task. Dealing with this particular issue is actually related to techniques such as task migration[24,39], redundant submissions for increased reliability[30], and so on. Such techniques constitute another line of work on their own and are thus out of scope of this paper.

In this paper, we propose the integration of resource discovery and scheduling for mobile ad hoc grids into a single protocol called DICHOTOMY (DIscovery and sCHeduling prOTOcol for MobilitY). This protocol allows the computational tasks that comprise an application to be distributed among the most resourceful nodes in a MANET. Such nodes are selected automatically by the protocol, based on some *suitability* criteria defined by the inquiring application. The suitability of a node is determined in terms of the resources the application is interested in and the relative importance among them from the viewpoint of the application. By always selecting the most resourceful nodes and making advance reservations of the resources, our proposed protocol provides an implicit scheduling mechanism that balances the load among resource providers in the mobile ad hoc grid. For computationally-intensive applications, such balance also helps with lowering the average completion time of tasks in the mobile ad hoc grid.

The DICHOTOMY protocol works in a peer-to-peer fashion, regardless of underlying routing protocols. The protocol can be classified as a purely query-based discovery protocol–requests are broadcast over the network on demand and providers reply to these requests accordingly. Query-based protocols are known to cause waste of resources if consumers naively broadcast requests by flooding (also referred to as the broadcast storm problem[37]) or if providers naively reply to such requests (a.k.a. the reply implosion problem[11,8]). To mitigate broadcast storms, a simple mechanism that limits the range of flooding is used. To avoid reply implosions, an in-network filtering algorithm is employed, which allows the more suitable replies to suppress unnecessary replies from other nodes alongside the paths used for forwarding the more suitable replies back to the inquiring node.

In our previous publications[29,16,15] we gave an overview of the architecture in which the DICHOTOMY protocol is employed and of preliminary versions of the protocol and its mechanisms. The present paper builds upon these, presenting an extended version of the work that focuses specially on the details of the protocol implementation and its performance evaluation. In particular, besides providing further details of the protocol implementation and design choices, we also present a novel performance evaluation study that thoroughly analyzes the DICHOTOMY protocol covering key aspects such as scheduling assessment, discovery efficiency under mobility, and scalability analysis. Our prototypical implementation–deployed both in an experimental ad hoc grid testbed and in the NCTUns simulation and emulation platform[46] was focused on showing the feasibility of our approach under operating conditions. We also devised a simpler simulation model over the ns-2 simulator[33], to evaluate the scalability of our approach.

The remainder of the paper is structured as follows. Section 2 provides some background on mobile ad hoc grids. Section 3 introduces the DICHOTOMY protocol as well as its main features and mechanisms. We present some relevant implementation details in Section 4. We evaluate the performance of our proposed protocol in Section 5. Section 6 discusses related work in discovery and scheduling in wireless and mobile ad hoc grids. Finally, in Section 7 we summarize the contributions of our proposed protocol and discuss possible future work.

2. Background

In this section, we discuss why the discovery and scheduling mechanisms have to be different for MANETs and present the basic ideas behind our proposed DICHOTOMY protocol that impact discovery and scheduling procedures. We also introduce the MoGrid architecture[29], our underlying mobile ad hoc grid middleware.

2.1. Resource discovery and scheduling in mobile ad hoc grids

To the best of our knowledge, the research on combined resource discovery and scheduling for mobile ad hoc grids is still rather preliminary. The mechanisms for resource discovery and scheduling in such grids must be fundamentally different from those used in wired grids as the former are much more sensitive to network behavior than the latter. This is due to the very dynamic nature of MANETs for instance, nodes may move, or the QoS properties of the wireless medium can vary over short periods of time. In MANETs, there are two ways for devices to exchange resource information[14]: *queries* and *announcements*. Queries involve some form of devices sending requests throughout the network and making the devices providing the required resource information reply to these requests. Announcements permit that

devices advertise their provided resources in the network to interested devices. Most MANET applications actually employ a combination of the two techniques. The scope of diffusion of queries and announcements as well as the periodicity and caching policies of announcements are important parameters that determine the efficiency and accuracy of resource discovery in such networks.

In the particular case of mobile ad hoc grids, announcements are inadequate for offering resource information. Besides the declaration of the availability of a resource becoming time-sensitive due to node mobility, the availability of *highly* dynamic resources, such as CPU load and available memory, may vary considerably in very short periods of time[10,3]. Therefore, we adopt a purely query-based approach in our proposed DICHOTOMY protocol (further details in Section 3). Moreover, we argue that discovery and scheduling must be approached in an integrated way in mobile ad hoc grids, in particular for multihop scenarios. This is due to two main reasons:

1. The discovery of highly dynamic resources should be associated with some form of advance reservation[42,48], so as to ensure that these resources are indeed available at the resource provider when the resource consumer submits the intended computational task.

2. The messaging overhead and the information required for discovering available (and reachable) resource providers in the mobile ad hoc grid can be used as well for automatically selecting the set of most appropriate providers. The automatic selection of resource providers bears resemblance to metascheduling techniques in traditional grids[7], and is particularly important for applications targeted at solving computationally-intensive problems.

As shown in Section 3, our proposed DICHOTOMY protocol implements advance reservation and an in-network filtering algorithm that allows the automatic selection of the most suitable resource providers.

2.2. MoGrid architecture

To support the establishment of mobile ad hoc grids, we proposed in Lima et al.[29] a middleware architecture, called MoGrid, which is depicted in Figure 1. In our work, we refer to a mobile ad hoc grid as a set of (possibly heterogeneous) mobile nodes in a MANET that manage resource usage and provisioning for applications in a decentralized way. Moreover, in our architecture resourceful nodes (e.g. laptops) may provide resources to multiple application tasks simultaneously, while other, resource-constrained nodes (e.g. mobile phones) may not provide resources at all in these cases, such nodes may only ask for resources from remote nodes. We further discuss the issue of node heterogeneity in Section 7.

In the remainder of this section we provide some background on this architecture, so as to identify some requirements on the design of the DICHOTOMY protocol.

The MoGrid architecture supports applications in two phases. First, nodes acting as resource consumers discover other nodes acting as resource providers by means of services that the discovery layer offers. The DICHOTOMY protocol is used for implementing these services. Second, resource consumers submit application tasks to selected resource providers according to the resources they make available by means of services the submission layer offers. Protocols within this layer are out of the scope of this paper; nevertheless, we assume a resource usage model in which a resource consumer has a *group* of independent tasks (e.g. bag of tasks) which it needs to submit for execution on resource providers. The adopted resource usage model implies the use of a mechanism during the discovery phase that selects *multiple* resource providers simultaneously. This is a major shortcoming of current discovery approaches to MANETs, as discussed in Section 6.

Each kind of application may have its specific resource requirements on the mobile ad hoc grid. As an example, for long-lived CPU-intensive applications battery power is more important than highly available bandwidth when submitting their tasks for execution, whereas storage capacity and connection stability have greater importance to data replication applications. Therefore, the selection mechanism must take into account the current level of resource usage associated with a given provider and the relative importance between its resources from the application's point of view.

The MoGrid architecture supports applications implemented directly on top of the discovery layer, or on top of an application-specific layer (see Figure 1). One of the purposes of the application-specific layers is to implement different resource weightings according to different application types, as for instance computationally-intensive, communication-intensive, or data intensive applications (Section 5 provides an example of implementation of an application-specific layer).

The focus of the MoGrid architecture on *resource* discovery instead of *service* discovery also implies the use of a query-based approach to resource discovery, as opposed to announcement-based approaches (see discussion on Section 6). This is because the highly-variable availability of resources such as CPU load and available memory may demand a higher announcement rate, which in turn may lead to an increased consumption of other resources in the MANET, such as network bandwidth and energy.
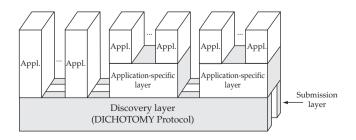

Figure 1. ModGrid arquitecture.

3. The DICHOTOMY Protocol

In this section, we present the DICHOTOMY protocol for integrated discovery and scheduling in mobile ad hoc grids. First, we discuss the basic operation of the proposed protocol (Section 3.1). Next (Section 3.2), we describe the operation of the adopted mechanism for determining the suitability of nodes (acting as resource providers) to answer particular requests from inquiring nodes (acting as resource consumers). Finally (Section 3.3), we describe how the protocol avoids the reply implosion problem.

3.1. Basic operation

The DICHOTOMY protocol defines two main messages: INITIATORREQUEST (IREQ) and COLLABORATORREPLY (CREP). An IReq message conveys:

i) A unique request identifier used to match requests to replies (REQID);

ii) The maximum reply delay that the inquiring node is willing to tolerate (MAXREPLYDELAY) c.f. Section 3.2;

iii) The number of resource providers to which the inquiring node wishes to submit tasks (NUMMAXREPLIES);

iv) Information about the resources the application is interested in and the relative importance among them we call this the *contextual information* (ctxtInfo) of the request;

v) The current and maximum diameter (in number of hops) of the request propagation (NUMHOPS and maxHops, respectively); and

vi) A unique identification of the last node (e.g. its MAC address) sending out the message (hopID). The CREP message will be explained later in this section.

Figure 2 illustrates the basic operation and involved message exchange within DICHOTOMY. An inquiring node sends out IREQ messages to the other nodes in the mobile ad hoc grid to ask them for resource provisioning. The IREQ message is replicated at each intermediate node to form a controlled flood limited by the maxHops parameter. This is

shown in Figure 2 as node i, the inquiring node, sends an IREQ message to node t, an ordinary intermediate node, that replicates the IREQ message to all its neighbors; the process is repeated until the maxHops value is achieved, making the IREQ message reach for instance nodes y and z.

Upon reception of an IREQ message, a node records it as a pending request in a local data structure (pendingList). In addition to REQID, numMaxReplies, and hopID, which are obtained from the IREQ message, each entry of pendingList has a numReplies field (initially set to 0) and two associated timers (replyDelay and cleanUp). replyDelay is described in Section 3.2; numReplies and cleanUp are described in Section 3.3.

After updating PENDINGLIST, the receiver node replies to an IREQ message depending on its willingness to collaborate as a resource provider, as described in Section 3.2. In addition to replying to requests, nodes may also forward requests to other nodes farther away from the inquiring node in the mobile ad hoc grid, if NUMHOPS < MAXHOPS. Such forwarding is done through simple flooding to neighboring nodes a scheme for inhibiting redundant rebroadcasts[37] should be employed in this case. Before being forwarded, a request has its numHops field incremented and its hopID field updated with the identification of the current forwarding node. Forwarding requests with updated hopIDs allows nodes farther away from the inquiring node to keep track (in their local pendingList structures) of the path traversed by the request, which will be used for determining the *return path* of the corresponding replies.

Nodes willing to collaborate as resource providers (hereafter, *collaborating nodes*) send CREP messages to inquiring nodes in response to IREQ messages. In Figure 2, it is assumed that all nodes in the mobile ad hoc grid collaborate by replying to the inquiring node. In particular, nodes t, y, and z send CREP messages as well as possibly other collaborating nodes whose CREP messages are represented by dots in Figure 2. A CREP message informs the inquiring node about the collaborating node address (identified by a collAddr field) as well as its resource availability according to the contextual information of interest indicated in the
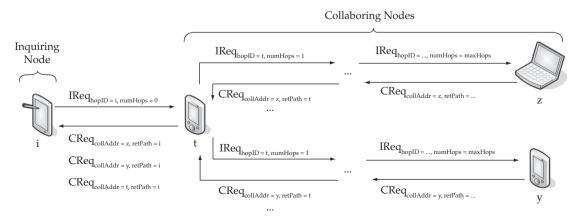

Figure 2. Example of DICHOTOMY messages.

corresponding request (resInfo field). Besides the collAddr and resInfo fields, a CREP message also conveys:

i)   The REQID matching that of the corresponding IREQ message; and

ii) The identification of the node from which the corresponding request was received (RETPATH)–the replying node obtains such an identification from the HOPID field in the corresponding IREQ message in the PENDINGLIST.

A collaborating node sends a CREP message towards the inquiring node through an application-level forwarding mechanism that uses the already known path traversed by the request. When a node receives a CREP message, it is processed according to Algorithm 1. If the reply corresponds to a request the node has previously originated, the node processes the message and prevents this message to be further forwarded in the network (lines 2 and 3). If the reply is instead addressed to an inquiring node other than the receiver, the latter first checks whether there is an entry for the corresponding request in its pendingList structure (lines 6 and 7). If not, the reply is silently discarded.[I] Otherwise, the receiving node determines whether this reply must be forwarded to the inquiring node (line 8), according to the algorithm described in Section 3.3. If this is the case, the node updates the reply's RETPATH field with the value of the hopID field stored in the corresponding entry of pendingList and forwards the reply (lines 9 and 10). This allows a neighbor node along the return path to also forward such a reply to the inquiring node.

The application-level forwarding mechanism for reply messages avoids the use of network-level routing protocols that would generate additional network load on the MANET during the discovery phase. Besides, with small modifications

---

I. If maxHops = ∞, this condition should not happen because every node in the mobile ad hoc grid would in principle receive all IReq messages. Nevertheless, when a scheme for inhibiting redundant rebroadcasts is used, it is not always guaranteed that an IReq message will reach all nodes in the network even for maxHops = ∞.

Algorithm 1 CREP message main processing

Input: *msg* /*Received CREP*/
1: **if** FIRSTCOPY (*msg*) **then**
2:      **if** MYREPLY (*msg*) **then**
3:          PROCESS (*msg*)
4:          **return**
5:      **end if**
6:      *entry* ← *pendingList*[*msg.reqID*]
7:      **if** *entry* ≠ *NULL* **then**
8:          **if** CANFORWARD (*entry, msg*) **then**
9:              *msg.retPath* ← *entry.hopID*
10:             FORWARD (*entry.hopID, msg*)
11:         **end if**
12:         **return**
13:     **end if**
14:     DISCARD (*msg*)
15: **else**
16:     .../* Deal with duplicate replies */
17: **end if**

---

in the hopID and retPath fields–instead of storing the last hop in the path, they should in this case store the whole sequence of traversed nodes, inquiring and collaborating nodes could also learn the entire path between each other, and then use source routing–a fairly common approach used in some routing protocols for MANETs, such as DSR[23] at the beginning of the task submission phase. Details on the mapping of the application-level forwarding mechanism for reply messages onto the link level are discussed in Section 4.3.

3.2. Determining suitability

In the DICHOTOMY protocol, every node willing to collaborate with the provision of a specific resource delays the transmission of its CREP message according to the REPLYDELAY timer in the corresponding entry of its PENDINGLIST. Such a timer is set so that more resourceful nodes reply earlier. This way, a node willing to collaborate can detect, before sending its own CREP message, whether other, more suitable nodes have already replied to the corresponding request. If the total number of replies generated in the mobile ad hoc grid is larger than the value N_M of the NUMMAXREPLIES field in the corresponding IReq message, and if the inquiring node chooses the N_M first received replies, the protocol guarantees that the inquiring node discovers the N_M most suitable nodes for providing the resource. Moreover, when used together with the algorithm presented in Section 3.3, the strategy of delaying replies helps in reducing the total number of replies being conveyed in the mobile ad hoc grid by eliminating unnecessary additional replies alongside the return path from the replying node to the inquiring one.

It should be remarked that the determination of reply delays is flexible with regard to the resources to be taken into account e.g. connectivity status, CPU load, available memory, remaining battery power and the relative importance among them. Information about resources to be considered when computing the suitability of a node is conveyed by IREQ messages in the CTXTINFO field. When a node receives a request, it gathers its current state in terms of the resources of interest to compute the reply delay. For this algorithm to work correctly, all nodes in the mobile ad hoc grid must use the same criterion for this computation. In our implementation (c.f. Section 4), a collaborating node sets the REPLYDELAY timer in the corresponding entry of its PENDINGLIST to τ units of time, as given by Equation 1:

$$\tau = \left(1 - \omega \sum_{i=1}^{N} \left(\frac{\alpha_i P_i}{\sum_{j=1}^{N} P_j} \right) \right)(D_{max} - 2HS) \qquad (1)$$

where $0 \leq \alpha \leq 1$ and $0 \leq \omega \leq 1$. N represents the number of different resource types the collaborating node should take into account. P_i is the weight that describes the relative importance of each resource type i from the viewpoint of the application on the inquiring node, $1 \leq i \leq N$. Both N and P_i are described as part of the CTXTINFO field in the request. D_{max} is the maximum reply delay, which is also obtained from the

request (maxReplyDelay field). α_i is the normalized level of current availability (in the interval [0,1]) of resource type i at the collaborating node. ω indicates the willingness (also in the interval [0,1]) of the collaborating node to participate in the resource provisioning. τ is undefined for $\omega = 0$; such a value means that no replies will be sent by the collaborating node Section 4.1 provides an example of ω usage in our implementation. Finally, H and S are used for considering the transfer delays that IREQ and CREP messages may experience. H is the distance in hops between the collaborating node and the inquiring node, and S is a tuning parameter representing the mean transfer delay at each hop.

3.3. Avoiding the reply implosion problem

According to Algorithm 2, CREP messages are selectively forwarded towards the inquiring node.

Whenever the function CANFORWARD() is called, it increments the value N_R of the numReplies field in the entry of pendingList corresponding to the received CREP message (line 1), *regardless of the receiving node being on the return path of the reply*. The node's identification is then compared with the value of the retPath field in the reply (line 2). If the values are equal, it means the receiver is in the return path of the reply. The receiving node, however, will only be able to forward the reply to the inquiring node if $N_R \leq N_M$, where N_M is the value of the numMaxReplies field in the corresponding entry of its pendingList (line 3). If $N_R > N_M$, the node suppresses the reply. Note that each intermediate node in the return path implicitly informs its own vicinity due to the application-level forwarding mechanism about requests that have already been replied, thus allowing for further suppressions at nodes nearby the reply's return path.

To allow further suppressions, an entry in pendingList is kept alive until its cleanUp timer expires. In our implementation, a node sets cleanUp for entries in pendingList connected with replies *not* originated from this node to $\tau_{max} = D_{max} - 2HS$ units of time. For entries associated with replies originated from the node (i.e. its resource offerings), cleanUp is set to $2\tau_{max}$, to take account of advance reservations (*c.f.* Section 4.1).

Figure 3 illustrates an example of the operation of Algorithm 2. In the initial configuration (Figure 3a), only nodes w and z are within y's transmission range. y receives a reply to a request with REQID = 1000, and increments the

value N_R of the numReplies field in the corresponding entry of its pendingList (Figure 3b). Since y is in the return path of the reply (Figure 3c), y forwards such a message towards w, which is y's next hop in the return path. z overhears this transmission due to the characteristics of the mapping of the application-level forwarding mechanism onto the link level one (discussed in Section 4.3) and then increments the value N_R of the numReplies field in the corresponding entry of its pendingList, but does not forward that reply because it is not in the reply's return path. z receives another reply to the request with REQID = 1000 (Figure 3d), but although it is in this reply's return path, it does not forward the message because the numMaxReplies field in the corresponding entry of its pendingList indicates that it has already either forwarded or overheard N_M messages.

4. Implementation

We have implemented the DICHOTOMY protocol in Java, as part of the implementation of our MoGrid middleware architecture[29,II] The implementation is done in J2SE according to the restrictions of the CDC (Connected Device Configuration) profile of the Java ME. CDC is a standards-based framework for building and delivering mobile applications that can be shared across a range of network-connected personal mobile devices, such as grid services as seen in mobile ad hoc grids discussed in this paper. Typically, these devices include a 32-bit microprocessor/controller and require about 2 MB of RAM and 2.5 MB of ROM for the Java application environment. Our implementation of the protocol uses a monitoring service available as part of the MoCA architecture. This service is responsible for gathering information about the current state of a mobile node, including connectivity, CPU load, available energy and memory, and disk storage space.

4.1. Setup of parameters

Parameters S and ω in Equation (1) are configurable in our implementation. For our experiments (*c.f.* Section 5) we set $S = 10$ ms which is of the same order of magnitude as 100 m one-trip packet delays for 1 Mbps transmission rate and 1500-byte packets (disregarding delay variations due to queuing and medium access contention). ω is computed as $\omega usr/\beta L$, where ω_{usr} describes the user's level of interest (in the interval [0,1]) in allowing its device to collaborate with others on the mobile ad hoc grid, L is the number of entries in a node's pendingList corresponding to this node's current resource offerings, and β is an scalar factor. The denominator βL implements a simple mechanism for advance reservation by decreasing the node's willingness to collaborate when the number of pending requests at this node increases; β in this case controls how fast the influence of a user's willingness is lowered as its pendingList grows. We set $\beta = 2$ arbitrarily in our implementation. Finetuning this factor was left for future work.

Algorithm 2 Function CANFORWARD ().

Input: *entry*/*Entry in PENDINGLIST */ and
    msg/*Received CREP */

Output: boolean /* CREP is to be forwarded? */

  1: *entry.N_R* $\leftarrow$ *entry* N_R + 1
  2: **if** *msg.retPath = localID* **then**
  3:     **if** *entry.N_R* $\leq$ entry.N_M **then**
  4:        **return true**
  5:     **end if**
  6: **end if**
  7: **return false**

---

z's pendingList

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	3	4	x		
...

y's pendingList

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	1	4	w		
...

(a) Initial configuration

z's pendingList

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	3	4	x		
...

y's pendingList

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	2	4	w		
...

(b) y receives a reply to a request

z's pendingList

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	4	4	x		
...

$CRep_{reqID = 1000, retPath = w}$

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	2	4	w		
...

(c) y rebroadcasts the reply towards w.

z's pendingList

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	4	4	x		
...

$CRep_{reqID = 1000, retPath = y}$
supressed by z
$N_R = N_M$

y's pendingList

reqID	N_R	N_M	hopID	τ	τ_{max}
1000	2	4	w		
...

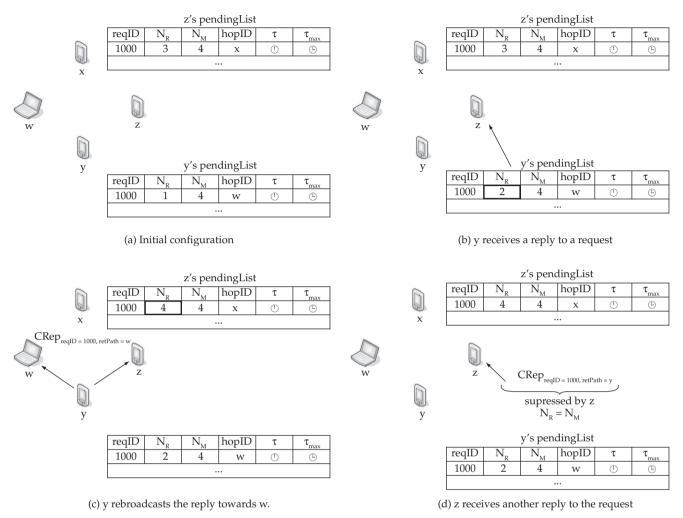(d) z receives another reply to the request

Figure 3. Scenario illustrating the suppression of CREP messages by Algorithm 2.

4.2. Resource description and matching

In the implementation of the DICHOTOMY protocol, the contextual information description conveyed by a request message (CTXTINFO field) and the matching of such description to the resource availability on the collaborating nodes are both based on simple attributes (key-value pairs). Semantic description languages e.g. ontology languages[32,34] could be also employed. Such languages support queries with more expressiveness and inference power, thereby allowing richer matching options (e.g. partial matchings). Such expressiveness, however, incurs some additional computational cost in terms of both processing time and memory footprint, which may be undesirable in scenarios involving resource-poor devices. The trade-off analysis of attribute and ontology-based resource description and matching for mobile ad hoc grids is out of the scope of this paper.

4.3. Application-level forwarding of reply messages

CREP messages are sent towards the inquiring node through an application-level forwarding mechanism. There are two alternative mappings of this scheme onto the link level: using unicast or broadcast/multicast transmissions.

For link-level unicast mappings, the RETPATH value associated with replies could be inferred from the destination address field in the encapsulating packets (*e.g.* the destination MAC address in IEEE 802.11 packets). This address field could convey the value of the HOPID field in the corresponding entry of PENDINGLIST which indicates the link-level address of the next node in the return path–as part of function FORWARD() (line 1 in Algorithm 1). By adopting such mappings, the retPath field carrying the link-level address of the next node in the return path could be omitted from the CREP messages, and the statement *msg.retPath←entry.hopID* (line 1) in Algorithm 1 could be removed. For a participating node to overhear replies from its neighbors, however, its network interface would have to work in promiscuous mode. Besides the security issues involved, this alternative has the drawback that, in promiscuous mode, the node must process the payload of *all* packets (not only those pertaining to the DICHOTOMY protocol) at the higher levels, which results in a waste of resources (CPU, memory and energy) that are crucial to computational tasks.

For link-level broadcast/multicast mappings, nodes do not need to set their network interfaces to work in promiscuous mode; however, the destination link address field in packets encapsulating reply messages do not specify a single recipient. Thus, the RETPATH field is necessary in such messages and line 1 in Algorithm 1 must be kept. Even so, we argue that the additional computational overhead of promiscuous mode operation introduced by unicast mappings can do more harm for applications targeted at solving computationally-intensive problems than the additional transmission overhead introduced by broadcast mappings. Therefore, we have adopted link-level broadcast mappings for the application-level forwarding mechanism in our implementation of the DICHOTOMY protocol.

It is worth noting that for MANETs in which the media access control is based on CSMA/CA (Carrier Sense Multiple Acess/ Collision Avoidance), broadcast transmissions are more unreliable and prone to collisions in comparison with unicast transmissions[37]. This is mainly due to the lack of acknowledgments, of RTS/CTS (Request/Clear to Send) dialogues, and of a mechanism for collision detection. The problem of collisions in link-level broadcast transmissions may be rather alleviated by the DICHOTOMY protocol, since replies from different resource providers are time-shifted, as discussed in Section 3.2. Regarding the lack of acknowledgments, an implicit acknowledgment mechanism for broadcast transmissions could be built upon the application-level forwarding mechanism. To understand this, consider again the example of Figure 3. When w receives the reply message from y (Figure 3c), w will forward the message because it is in the return path. Such a transmission will be overheard by y (since it is within w's range); y could then regard this transmission as a higher-level acknowledgement from w. Nonetheless, many subtle issues arise if a retransmission policy based on such implicit acknowledgments is devised to improve the reliability of the protocol. We argue that such additional complexity is not worthwhile, since CREP messages are always subject to suppression along the remaining path towards an inquiring node. Moreover, retransmissions add delays that may render the retransmitted CREP message useless at the inquiring node if it arrives after the predefined maxReplyDelay. In fact, the experimental results presented in Section 5.2 demonstrate that the DICHOTOMY protocol is efficient even without such a retransmission policy.

5. Performance Evaluation

We carried out our experiments in testbed and simulation environments, each of them to evaluate a different aspect of the DICHOTOMY protocol. These are described in the following subsections.

5.1. Scheduling Assessment

To assess the scheduling properties of the DICHOTOMY protocol under an increasing volume of requests, we set up an experimental ad hoc grid testbed. This testbed comprises 6 fixed, homogeneous nodes running Linux. These nodes have been arranged in a variety of multihop scenarios, with one node in each of such scenarios being used as the resource consumer and the remaining nodes as resource providers. Figure 4 illustrates the scenario from which the results presented in this section have been obtained. Other scenarios provided similar results, which have been omitted for brevity.

For our experiments, we implemented a simple master-worker matrix-matrix multiplication application on top of the testbed. To easily split the application into independent tasks, we employed a very simple distributed multiplication algorithm: given matrices $A_{m \times n}$ and $A_{n \times p}$, a master node computes $C_{m \times p} = AB$ by selecting p worker nodes with the DICHOTOMY protocol and sending to each worker node i ($1 \le i \le p$) a copy of matrix A along with $b_{n \times 1}^i$, i.e. the transposed vector whose elements are those of the i-th column of B. Each worker node i runs a task that computes matrix $c_{n \times 1}^i = Ab_{n \times 1}^i$ and returns it to the master node, which then builds the i-th column of C from $c_{n \times 1}^i$. The selection of the worker nodes in the experimental ad hoc grid that will run the tasks is made by only considering those nodes with the most available CPU and memory resources. We implemented an application-specific layer for this purpose (c.f. Section 2.2), with $N = 2$, $P_{CPU} = 4$, and $P_{mem} = 1$ in Equation 1.

For each testbed scenario, we ran a total of 30 experiments. In each experiment, the resource consumer sent a total of 30 IREQ messages at regular intervals of 30 s and set MAXREPLYDELAY = 10 s. Each resource provider s selected by a specific IREQ message was sent a task to compute $c_{n \times 1}^s$. Matrices $A_{m \times n}$ and $B_{n \times p}$ were dimensioned in such a way that such computation could generate a cumulative load on the resource providers. Figure 5 shows the cumulative load share for each node during one of these experiments. The other experiments on the testbed scenario depicted in Figure 4 provided similar results. We observe in the figure that the slave nodes have their loads balanced after the transient state (first 10 requests). This shows that our protocol performed an efficient and dynamic load balancing between resource providers under an increasing volume of requests.
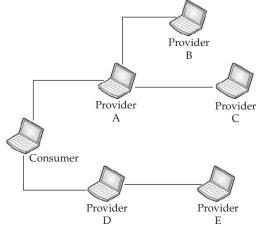

Figure 4. Testbed scenario.

We also used the testbed experiments described above to analyze the scheduling quality that our protocol provides to applications. We calculated the cumulative wall clock time (the time difference between the start and finish of a task) of all tasks generated in the testbed application after the 30 experiments with our protocol. We then compared it with the cumulative wall clock time achieved after 30 experiments by a scheduler that selects worker nodes randomly. As can be seen in Figure 6, our protocol schedules tasks among worker nodes in such a way that the cumulative wall clock time is approximately 36.5% smaller than that achieved by a random scheduler. Further, the time overhead implied by our protocol for discovery and scheduling, which is upper bounded by the MAXREPLYDELAY parameter (10 s in our simulation set up), is negligible with regard to the obtained cumulative wall clock time of the tasks, which is in the order of hundreds of thousands of seconds. These results indicate that our protocol also has a positive impact on the application efficiency.
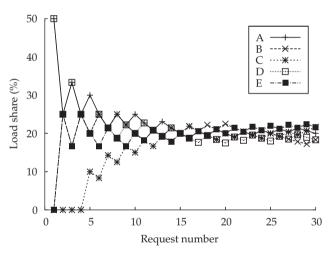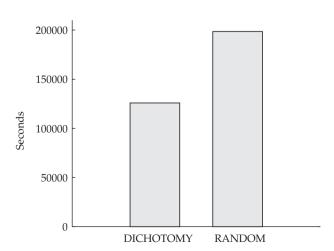
5.2. Discovery efficiency under mobility

The simulation scenarios consisted of 40 nodes placed in an obstacle-free, 500×500 m area. The initial position of each node in a specific simulation was set randomly, with the constraints that at the beginning of the simulation the nodes formed a connected topology, and the distance among the nodes was set to between 50 and 90% of the transmission range.

The first scenario consisted of a stationary topology. In the remaining scenarios, the movement of nodes followed the random walk model[4]. In such a model, each node moves in a random direction for some seconds–in a speed that is uniformly distributed in the range $]0, S_{max}] \times$ then chooses a new random direction, with no pauses in between such changes of direction. Table 1 summarizes the parameters adopted in the scenarios simulated with the NCTUns platform.

The discovery efficiency for each simulation scenario was measured as a sample proportion calculated over 100 runs. Each run consisted of a single resource consumer issuing a single IReq message throughout the simulated MANET. The sample proportion indicates the percentage of runs in which the protocol delivered *at least R* replies from resource providers to the resource consumer, as determined by the NUMMAXREPLIES field in the IREQ message. The number of collaborating nodes at each run was fixed at 10, which corresponds to 25% of the nodes in the simulated scenarios. Such a percentage was chosen based on the study by Hughes et al.[19], which states that in Gnutella–a famous P2P, collaboration-based file-sharing system–this percentage of participants is responsible for 98% of all service provisions.

Figure 7 presents the discovery efficiency of the DICHOTOMY protocol as a function of the maximum node

Figure 5. Cumulative load of slave nodes.

Table 1. Parameters for the NCTUns simulations.

Parameter	Value
Number of nodes (N)	40
Number of resource providers	10
Maximum number of replies (R)	4
Transmission range	100 m
Maximum node speed (S_{max})	0 to 5 m/s

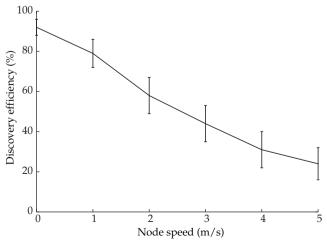**Figure 6**. Cumulative wall clock time of tasks.

Figure 7. Discovery efficiency in a mobile scenario.

speed (S_{max}). The vertical error bars correspond to the 95% confidence intervals for each sample proportion. The results show that the protocol behaves well under situations of human mobility (from 0.8 to 1.2 m/s, which corresponds to a typical range for pedestrian walking speeds[40]).

As can be observed in Figure 7, even for the stationary scenario ($S_{max} = 0$) the protocol does not reach 100% efficiency–the sample proportion is 92%, with ±4.13 confidence intervals. This is due to the drawbacks stated in Section 4.3 as regards the application-level forwarding mechanism for reply messages being mapped onto link-level broadcast transmissions in CSMA/CA enabled nodes.

5.3. Scalability Analysis

In principle, we could use our implementation of the DICHOTOMY protocol on the NCTUns platform to evaluate its scalability in scenarios with an increasing number of nodes. Nevertheless, any simulation of the DICHOTOMY protocol on the NCTUns platform was restricted to a maximum of 50 nodes; above these values the simulations generated a load beyond the capacity of our NCTUns simulation infrastructure. To allow a more extensive analysis of the scalability of the DICHOTOMY protocol, we devised a simpler simulation model over the ns-2 simulator[33]. Our experiments in such simulator considered fixed nodes forming topologies with a constant number of nodes within the same transmission range (node density), so that the impact of increasing the number of nodes in the ad hoc grid could be properly evaluated. The results presented in this section correspond to the average of 100 sample runs per simulated scenario with a 95% confidence level. This analysis was mainly focused on the evaluation of two metrics: the network load in the ad hoc grid due to reply messages, and the suppression diameter of these messages–the distance (in number of hops) between the inquiring node and the nodes where these messages were suppressed. Table 2 presents the parameters adopted in the scenarios simulated by ns-2.

The average network load in the ad hoc grid due to reply messages was computed using, for each scenario, the mean number of packets involving these messages.
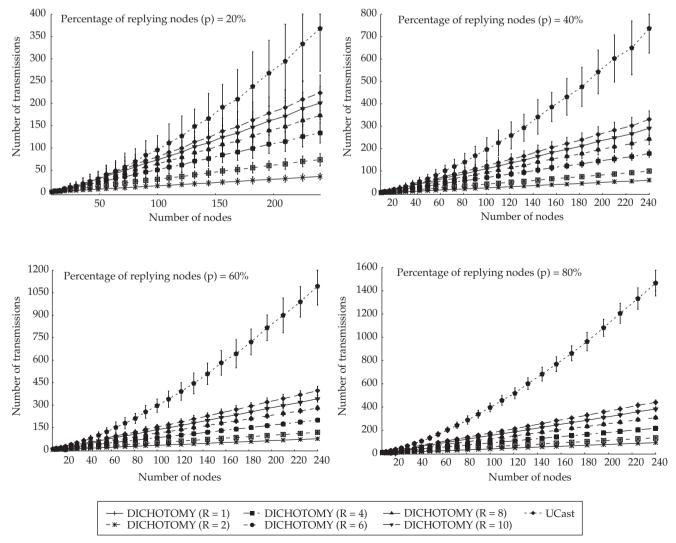

Figure 8. Network load in the mobile ad hoc grid due to reply messages.

Table 2. Parameters for ns-2 simulations.

Parameter	Value
Number of nodes (N)	10 to 240
Percentage of resource providers (p)	20% to 80%
Maximum number of replies (R)	1 to 10
Node density	5
Transmission range	12.5 m
Distance between nodes	10 m

Importantly, this metric also allow us to imply whether there is a significant reduction in energy consumption of devices in a mobile ad hoc grid due to the suppression of replies, given that transmissions are known to be responsible for a high energy consumption. Using this metric, we compared the DICHOTOMY protocol with a hypothetical query-based discovery protocol in which service replies are sent by unicast directly to inquiring nodes (we call it ``UCast'') in both protocols the inquiring nodes broadcast requests by flooding, and no service announcements are employed. In our belief, a comparison with advertisement- and hybrid-based approaches would not make sense for mobile ad hoc grids because the typical resources involved (e.g. CPU, memory) are highly dynamic (c.f. Section 2).

Figure 8 presents the network load due to reply messages as a function of the number of nodes for different percentages of nodes willing to collaborate as resource providers. The vertical error bars indicate the confidence intervals. The results show that the adoption of the DICHOTOMY protocol allows for an increasing reduction–with respect to the ``UCast'' protocol–in the total number of transmissions, as the number of devices in the mobile ad hoc grid increases. We also observe an even higher level of suppressions when there is a larger percentage of nodes (p) in the mobile ad hoc grid with interest in collaborating on resource provisioning. These results suggest the scalability of our approach.

The suppression diameter of reply messages allow us to evaluate the degree of distribution of the mitigation of the forwarding burden provided by the DICHOTOMY protocol among the nodes in a mobile ad hoc grid, and consequently the distribution of energy savings among such nodes due to the reduction in the amount of transmissions. Figures 9 and 10 present the distribution of suppressions as a cumulative distribution function (CDF) for different numbers of nodes and percentages of replying nodes. To better illustrate the distribution of suppressions through the network, the results presented in these figures are contrasted with a uniform CDF (represented by the straight line in the figures).

Comparing all graphs in Figures 9 and 10, we observe a better distribution of suppressions as the number of nodes and the percentage of replying nodes (p) increase. Again, this suggests the scalability of our proposed approach.

Comparing the curves of each individual graph, it is also possible to verify that the distribution of the suppression diameter becomes less uniform as the maximum number of replies (R) increases, with a tendency to suppressions being concentrated closer to the inquiring node. This is an expected behavior, since less suppressions occur at nodes farther away from the inquiring node and replies follow convergent paths towards such node.

6. Related Work

In spite of some research efforts over the past few years on wireless and mobile ad hoc grids, none of it as far as we can see has addressed the issues related to the interplay of resource discovery and scheduling in such grids. In the following we survey relevant research related to these two areas.

6.1. Resource discovery

Although some papers point out the need for resource discovery protocols in wireless grids[31,35], no work has explicitly addressed this area for multihop mobile ad hoc grids. A close subject–service discovery protocols (SDPs)–has been a hot topic in the area of MANET research[44] and could arguably be extended or adapted for resource discovery purposes. Nonetheless, most of the SDPs for MANETs are based on announcements, allowing nodes to advertise services in the network; nodes interested in such services cache the related advertisements. Clearly, such protocols are inadequate for offering computational services that depend on *highly* dynamic resources, such as CPU load and available memory. The availability of such resources may vary considerably in short periods of time[10,3], thus demanding frequent announcements, which in turn may lead to an increased consumption of other resources in the MANET, such as network bandwidth and energy. In this context, some pieces of work propose improvements to the broadcasting of service requests in multihop MANETs, so as to reduce the amount of packet transmissions related to such requests. Examples include Konark[27], Group-based Service Discovery (GSD)[5], and Field Theoretic Approach (FTA)[28].

Konark introduces the concept of *service gossiping*, in which a node can selectively forward both service requests and replies based on cached announcements from other nodes. The efficiency of the Konark approach, however, is highly dependent on caching of service information, thus being inadequate for grid-like computational services. The GSD architecture controls request broadcasts based on the semantic grouping of services as ontology classes, but its efficiency is also dependent on the advertisement and caching of such classes. When compared with the two previous approaches, FTA further reduces the amount of transmissions related to request forwarding by adopting an analogy of electrostatic fields. In the FTA approach, an inquiring node sends out a service request (a negative test charge), which is ``attracted'' by the most appropriate service instance (the positive charge that creates the field with highest potential). This way, FTA allows the automatic selection of a single node as the most suitable provider of a particular service, with
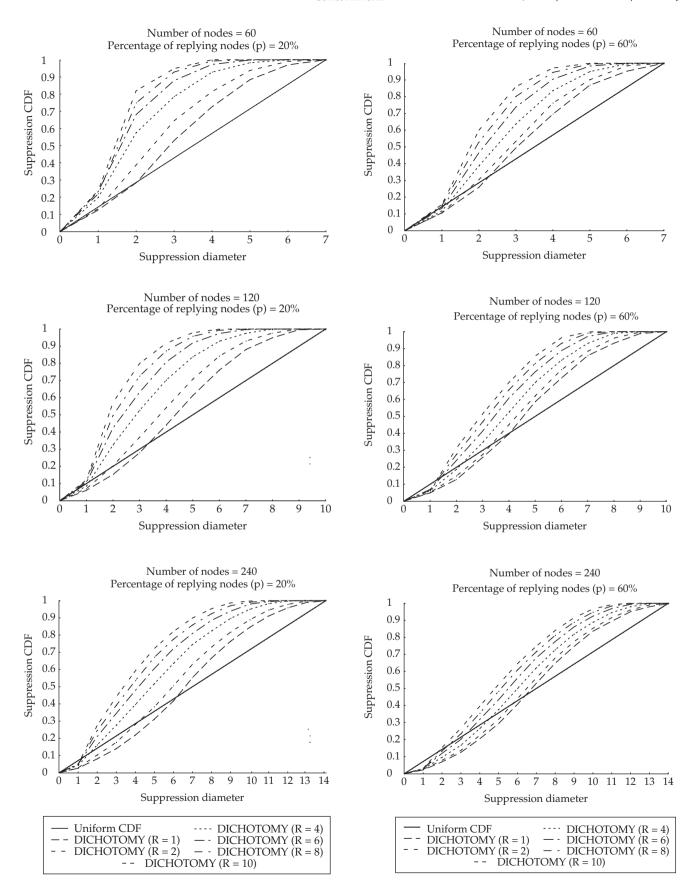
Figure 9. Distribution of reply suppressions (20% of replying nodes in the mobile ad hoc grid).

Figure 10. Distribution of reply suppressions (60% of replying nodes in the mobile ad hoc grid).

smaller overhead on the MANET as compared with Konark and GSD. Nevertheless, this approach disregards the automatic selection of multiple providers, and its efficiency is also highly dependent on the spread of potentials, which is based on periodic announcements.

Other researchers have focused on extending routing protocols to accomodate service discovery. One example comes from Varshavsky et al.[43] and their cross-layer approach to integrating service discovery functionality within the DSR[22] (query-based) and DSDV[38] (announcement-based) routing protocols. This approach and others similar to it[26,6] is also inappropriate for mobile ad hoc grids, since either its discovery efficiency is again dependent on caching of announced service information, or all replies from service providers matching the service description of a client's query arrive at the inquiring node for local selection by the client, thus wasting communication and energy resources.

6.2. Resource Scheduling

As far as traditional grids are concerned, many different scheduling strategies have been developed along the years[1,12,47]. Only very recently, however, have specific scheduling strategies for wireless and mobile ad hoc grids emerged in the literature[17,25,30,20,49]. Most of them employ task replication to optimize some objective function related to the inherent limitations in processing, memory, battery power, and wireless communications capabilities of mobile devices.

Huang et al.[17] propose a two-level scheduling model suitable for wireless grids. The first level of their model is responsible for mapping tasks to fixed grid nodes; some of these nodes act as proxies between a wireless domain (e.g. an area covered by an access point) and the fixed grid. The second level conducts scheduling within each proxy-centric wireless domain. The proxy runs a revised Min-Min heuristic algorithm, which aims at minimizing energy consumption at the mobile nodes. A similar approach based on hierarchical scheduling is proposed by Katzaros and Polyzos[25], with the difference that task replication is employed to treat disconnection events. Both approaches are inadequate for mobile ad hoc grids due to the need for a fixed node acting as the scheduler.

Litke et al.[30] focus on shortening the overall system response time with a scheme for task replication based on the knapsack problem formulation. Their strategy aims at maximizing the utilization of computational resources provided by the mobile nodes. A probability function is used for computing the availability of each node based on its current *failure rate* (e.g. number of times it was unreachable in the network), so that a node only receives tasks that may be completed before its mean time to failure (MTTF). Nevertheless, Litke et al. disregard energy consumption, which–in the case of multihop mobile ad hoc grids–may limit the operational life time of the whole system.

Zong et al.[49] aim at reducing schedule lengths of precedence-constrained parallel tasks while conserving energy at the mobile nodes. Their strategy relies on judiciously replicating tasks so that energy consumption related to inter-task communication is diminished; however, the process of node selection, as described in Wolf et al.[49], implies a centralized discovery service.

Hummel and Jelleschitz[20] propose a decentralized scheduler, thus bearing more resemblance to our approach; each mobile node decides autonomously whether to process a submitted task depending on its current and estimated near future capabilities. Their main focus, however, is on providing fault tolerance through task replication, which is coordinated by means of task queues managed in a distributed virtual shared memory. Crucially, their solution does not take into account the waste of communication and energy resources due to this coordination.

7. Summary and Outlook

In this paper, we have presented the specification, implementation, and performance evaluation of a novel protocol (DICHOTOMY) for integrated resource discovery and scheduling on multihop mobile ad hoc grids. Overall, our experimental results show that:

- Our protocol does appropriate resource scheduling, allowing an efficient load balancing among resource providers and having a positive impact on the application efficiency under an increasing volume of discovery requests;

- The discovery efficiency–i.e. the percentage of successfully answered requests–is kept at acceptable levels in the mobile application scenarios we are interested in, which involves pedestrian (walking) mobility;

- The proposed protocol scales very well with respect to an increasing number of nodes in comparison with the traditional query-based solutions for service discovery, since it increases the total amount of energy savings due to packet transmissions;

- The suppression of replies performed by our in-network filtering algorithm is not concentrated at specific points of a MANET–instead, the filtering is distributed among the nodes. As a consequence of the distributed reduction in the amount of transmissions, the energy savings due to nodes having fewer replies to retransmit are also distributed throughout the network.

During the development of this work, some aspects have been identified for future investigation:

- A more elaborated mechanism for advance reservations may be devised, including the treatment of resource ``underbookings'' a situation that might happen if the collaborating node reserves resources that are not actually used because its reply is suppressed before arriving at the inquiring node. Although such a reservation is temporary (as defined by the CLEANUP timer), it may affect concurrent requests.

- The MAXREPLYDELAY parameter has an important impact on the efficiency of the DICHOTOMY protocol. Finetuning this parameter e.g. as a func-

tion of the transmission delay of messages is essential to increase the discovery efficiency under mobility (since messages typically take no more than a few hundred milliseconds for delivery), and also to reduce the discovery time without increasing the number of reply collisions (which is achieved through the asynchronism in the transmission of these messages).

- The investigation of heterogeneity-aware suitability criteria is an interesting open issue. In this context, the theoretical and simulation study by Huang et al.[18] has recently brought interesting insights into how to handle nodes with heterogeneous computational power.

- Finally, in our current implementation the suitability of a resource provider is only computed by its own context (i.e. the state of its own resources) and not the context of intermediate nodes in its path to the inquiring node. This did not affect our results in the testbed because it formed topologies with just few hops. We are currently evaluating to what extent disregarding the context of intermediate nodes may affect a mobile ad hoc grid and investigating alternative solutions.

Acknowledgements

References

1. Abramson D, Sosic R, Giddy J and Hall B. Nimrod: a tool for performing parametised simulations using distributed workstations. In: *Proceedings of the 4 IEEE Symposium on High Performance Distributed Computing*; Washington, DC, USA. New York, NY, USA: IEEE Press; 1995.

2. Berman F, Fox G and Hey AJG. *Grid computing*: making the global infrastructure a reality. New York: John Wiley & Sons; 2003.

3. Bolosky WJ, Douceur JR, Ely D and Theimer M. Feasibility of a serverless distributed file system deployed on an existing set of desktop PCs. *SIGMETRICS Performance Evaluation Review* 2000; 28(1):34-43.

4. Camp T, Boleng J and Davies V. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications* 2002; 2(5):483-502.

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

6. Cheng L and Marsic I. Service discovery and invocation for mobile ad hoc networked appliances. In: *Proceedings of the 2 International Workshop on Networked Appliances*; 2000; New Brunswick. USA; 2000.

7. Czajkowski K, Foster I and Kesselman C. Resource and service management. In: Foster I and Kesselman C. (Eds.). *The grid* 2: blueprint for a new computing infrastructure. 2 ed. New York: Morgan Kaufmann Publishers; 2003. p. 259-284. (cap. 18)

8. Dabrowski C, Mills KL and Quirolgico S. *A model-based analysis of first-generation service discovery systems*. Gaithersburg, MD, USA: National Institute of Standards and Technology; 2005. (Technical report, NIST special publication 500-260)

9. Denning PJ. Hastily formed networks. *Communications of the ACM* 2006; 49(4):15- 20.

10. Dinda PA. The statistical properties of host load. *Scientific Programming* 1999; 7(3-4):211-229.

11. Erramilli A and Singh RP. A reliable and efficient multicast for broadband broadcast networks. *ACM SIGCOMM Computer Communication Review* 1987; 17(5):343-352.

12. Frey J, Tannenbaum T, Foster I, Livny M and Tuecke S. Condor-G: a computation management agent for multi-institutional grids. *Cluster Computing* 2002; 5(3):237-246.

13. Gaynor M, Moulton SL, Welsh M, LaCombe E, Rowan A and Wynne J. Integratng wireless sensor networks with the grid. *IEEE Internet Computing* 2004; 8(4):32-39.

14. Gomes ATA, Ziviani A, Lima LS and Endler M. Service discovery approaches to mobile peer-to-peer computing. In: Seet BC. (Ed.). *Mobile peer-to-peer computing for next generation distributed environments*: advancing conceptual and algorithmic applications. Hershey, PA, USA: IGI Gobal; 2009. p. 437-459. (cap. 20)

15. Gomes ATA, Ziviani A, Lima LS, Endler M and Chelius G. Mitigating reply implosions in query-based service discovery protocols for mobile wireless ad hoc networks. In: *Proceedings of the 7 International Conference on Ad-hoc, Mobile & Wireless Networks*; 2008; Sophia Antipolis, France. Heidelberg, Germany: Springer; 2008. p. 29-42.

16. Gomes ATA, Ziviani A, Lima LS and Endler M. DICHOTOMY: A resource discovery and scheduling protocol for multihop ad hoc mobile grids. In: *Proceedings of the 1 IEEE InternationalWorkshop on Context Awareness and Mobility for Grid Computing*; 2007; Rio de Janeiro, RJ, Brazil.

17. Huang CQ, Zhu ZT, Wu YH and Xiao ZH. Power-aware hierarchical scheduling with respect to resource intermittence in wireless grids. In: *Proceedings of the 5 IEEE International Conference on Machine Learning and Cybernetics*; 2006; Dalian, China. New York, NY, USA: IEEE Press; 2006. p. 693-698.

18. Huang J and Lee SY. A heterogeneityaware approach to load balancing of computational tasks: a theoretical and simulation study. *Cluster Computing* 2007; 11(2):133-149.

19. Hughes D, Coulson G and Walkerdine J. Free riding on gnutella revisited: the bell tolls? *Distributed Systems Online* 2005; 6(6).

20. Hummel KA and Jelleschitz G. A robust decentralized job scheduling approach for mobile peers in ad-hoc grids. In: *Proceedings of the 7 IEEE International Symposium on Cluster Computing and the Grid*; 2007; Rio de Janeiro, RJ, Brazil. New York, NY, USA: IEEE Press; 2007. p. 461-471.

21. Hwang J and Aravamudham P. Middleware services for P2P computing in wireless grid networks. *IEEE Internet Computing* 2004; 8(4):40-46.

22. Johnson D, Maltz D and Broch J. DSR: the dynamic source routing protocol for multihop wireless ad hoc networks. In: Perkins CE. (Ed.). *Ad Hoc Networking*. Upper Saddle River, NJ, USA: Addison-Wesley; 2001. p. 139-172. (cap. 5)

23. Johnson D, Maltz D and Hu YC. *The Dynamic Source Routing Protocol (DSR) for mobile ad hoc networks for ipv4*. California: IETF; 2007. (RFC 4728, experimental)

24. Kagawa K, Yamada K, Kamiya T and Nagata M. Fault tolerant grid migration using network storage. In: *Proceedings of Parallel and Distributed Processing Techniques and Applications*; 2006; Las Vegas, NV, USA. Bogart, GA, USA: CSREA Press; 2006. p. 241-245.

25. Katsaros K and Polyzos GC. Optimizing operation of a hierarchical campus-widemobile grid for intermittent wireless connectivity. In: *Proceedings of the 15 IEEE Workshop on Local and Metropolitan Area Networks*; 2007; Princeton, NJ, USA. New York, NY, USA: IEEE Press; 2007. p. 111-116.

26. Koodli R and Perkins C. *Service discovery in ondemand ad hoc networks*. California: IETF; 2000. (Expired)

27. Lee C, Helal A, Desai N, Verma V and Arslan B. Konark: a system and protocols for device independent, peer-to-peer discovery and delivery of mobile services. *IEEE Transactions on Systems, Man and Cybernetics* 2003; 33(6):682-696.

28. Lenders V, May M and Plattner B. Service discovery in mobile ad hoc networks: A field theoretic approach. *Pervasive and Mobile Computing* 2005; 1(3):343-370.

29. Lima LS, Gomes ATA, Ziviani A, Endler M, Soares LFG and Schulze BR. Peer-to-peer resource discovery in mobile grids. In: *Proceedings of the 3 International Workshop on Middleware for Grid Computing*; 2005; Grenoble, France.

30. Litke A, Skoutas D, Tserpes K and Varvarigou T. Efficient task replication and management for adaptive fault tolerance in mobile grid environments. *Future Generation Computer Systems* 2007; 23(2):163-178.

31. Litke A, Skoutas D and Varvarigou T. Mobile grid computing: changes and challenges of resource management in a mobile grid environment. In: *Proceedings of International Conference on Practical Aspects of Knowledge Management*; 2004; Viena, Austria.

32. Defense Advanced Research Projects Agency - DARPA. *DAML services*. Arlington: DARPA; 2006.

33. Information Sciences Institute - ISI. *The network simulator ns-2*. California: ISI; 1995.

34. World Wide Web Consortium - W3C. *Ontology Web Language reference*. Cambridge, MA, USA: W3C; 2004.

35. McKnight LW and Howison J. Towards a sharing protocol for wireless grids. In: *Proceedings of International Conference on Computing, Communication and Control Technologies*; 2003; Orlando, FL, USA.

36. McKnight LW, Howison J and Bradner S. Wireless grids: distribute resource sharing by mobile, nomadic, and fixed devices. *IEEE Internet Computing* 2004; 8(4):24-31.

37. Ni SY, Tseng YC, Chen YS and Sheu JP. The broadcast storm problem in a mobile ad-hoc network. In: *Proceedings of the 5 Annual ACM/IEEE International Conference on Mobile Computing and Networking*; 1999; Seattle, WA, USA. New York, NY, USA: IEEE Press; 1999. p. 151-162.

38. Perkins CE and Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Communication Review* 1994; 24(4):234-244.

39. Priya SB, Prakash M and Dhawan KK. Fault tolerance-genetic algorithm for grid task scheduling using check point. In: *Proceedings of the 6 International Conference on Grid and Cooperative Computing*; 2007; Washington. Washington: IEEE Computer Society; 2007. p. 676-680.

40. Rouphail N, Hummer J, Milazzo J and Allen P. *Literature review for Chapter 13, Pedestrians, of the Highway Capacity Manual*. Washington: Federal Highway Administration; 1998. (Technical report)

41. Sacramento V, Endler M, Rubinsztejn HK, Lima LS, Goncalves K, Nascimento FN et al. MoCA: a middleware for developing collaborative applications for mobile users. *IEEE Distributed Systems Online* 2004; 5(10).

42. Smith W, Foster I and Taylor V. Scheduling with advance reservations. In: *Proceedings of the 14 International Parallel and Distributed Processing Symposium*; 2000; Cancun, Mexico. New York, NY, USA: IEEE Press; 2000. p. 127-132.

43. Varshavsky A, Reid B and Lara E. A crosslayer approach to service discovery and selection in MANETs. In: *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*; 2005; Washington, DC, USA. New York, NY, USA: IEEE Press; 2005. p. 8.

44. Ververidis CN and Polyzos GC. Service discovery for mobile ad hoc networks: A survey of issues and techniques. *Communications Surveys & Tutorials* 2008; 10(3):30-45.

45. Waldburger M and Stiller B. Toward the mobile grid: Service provisioning in a mobile dynamic virtual organization. In: *Proceedings of IEEE International Conference on Computer Systems and Applications*; 2006; Dubai/Sharjah, UAE. New York, NY, USA: IEEE Press; 2006. p. 579-583.

46. Wang SY, Chou CL, Huang CH, Hwang CC, Yang ZM, Chiou CC et al. The design and implementation of the NCTUns 1.0 network simulator. *Computer Networks* 2003; 42(2):175-197.

47. Weng C and Lu X. Heuristic scheduling for bag of tasks applications in combinationwith QoS in the computational grid. *Future Generation Computing Systems* 2005; 21(2):271-280.

48. Wolf L, Delgrossi L, Steinmetz R, Schaller S and Wittig H. Issues of reserving resources in advance. In: *Proceedings of ACM International Workshop on Network and Operating System Support for Digital Audio and Video*; 1995; Durham, NH, USA. New York, NY, USA: ACM Press; 1995. p. 151-162.

49. Zong Z, Nijim M, Manzanares and Qin X. Energy efficient scheduling for parallel applications on mobile clusters. *Cluster Computing* 2008; 11(1):91-113.