Sharing Hardware Resources in Heterogeneous Computer-Supported Collaboration Scenarios

Davide Vega[a], Roc Messeguer[a], Sergio F. Ochoa[b], José A. Pino[b], Felix Freitag[a], Esunly Medina[a], Dolors Royo[a]
[a] *Department of Computer Architecture, Universitat Politècnica de Catalunya, D6 Building, Jordi Girona 1-3, 08034 Barcelona, Spain*
[b]*Department of Computer Science, Universidad de Chile, Av. Blanco Encalada 2120, Santiago, Chile*

Abstract. There currently are many mobile computing devices with various properties and capabilities. These devices may need to collaborate among them to allow nomad workers to perform a common activity. Unfortunately software developers in charge of creating infrastructures or applications allowing these devices to cooperate among them, do not count with clear guidelines to design such software components; particularly when these components must work in a scenario involving heterogeneous devices. This paper presents a study that tries to understand how to address collaboration among heterogeneous mobile devices, by exploring several variables affecting the process. In particular, this study explores various strategies to borrow CPU slots from peer mobile computing devices and return the favor back later on. The study outcomes indicate there is a short list of computing and network variables affecting the collaboration capability of the mobile devices. These findings have been verified using data mining techniques. Based on these findings and the lessons learned, the article presents a simulation method of computing scenarios that can help developers to determine which computing configuration would be suitable to be used in each particular work scenario. Software designers can take advantage of this simulation method and the design guidelines reported in this paper in order to develop applications able to work appropriately in heterogeneous computing scenarios.

Keywords: Hardware resources sharing, mobile collaboration, simulation method, heterogeneous scenario, tit-for-tat strategy.

1. Introduction

The computing and communication capabilities of mobile computing devices improve everyday. The cost reduction of these devices makes them accessible to most people, and their use opens new opportunities to perform computer-supported mobile collaboration. This collaboration type involves nomad users with mobile devices and a software system to perform on-demand interactions [32]. For instance, the interactions among medical personnel at a hospital [27] or the incidents discussion conducted by construction inspectors after reviewing building facilities [30]. Typically small devices (e.g. handhelds) are well prepared to support tasks involving high mobility [12]; however they are the most critical resource in these scenarios. Despite improvements on their performance, they still have significant hardware limitations (e.g. in computing power and memory capacity) to conduct non-trivial collaboration activities. These limitations could influence a user to not perform a specific task. Alternatively, if the user performs the activity without having the right hardware resources, then the activity could fail and the participants would get disappointed even in the case the collaborative system provides outstanding services. Thus, the success of a mobile collaboration is also dependent on the characteristics of the supporting devices [12]. The lack of guidelines to deal with this challenge pushes developers of these solutions to constrain the type of devices that can be used to run a certain application.

The approach proposed in this article is an alternative to that solution. We assume the work scenario is naturally heterogeneous in terms of computing devices, and the key is to take advantage of that fact when designing mobile collaborative applications.

Trying to find a solution for managing the devices heterogeneity and the hardware resources limitation of handhelds, we have seen the opportunity to use the volunteer [40] and public-resource computing concepts [3] for encouraging users to share their hardware resources. Thus, it is possible to perform non-trivial collaboration processes among mobile workers, even including those with less hardware resources.

The proposed solution creates decentralized collaboration networks, where mobile nodes (i.e. the device of a mobile worker) can share part of their hardware resources (e.g. processing power) during a given time to help other nodes. Thus, a user requiring extra hardware resources to perform a collaborative activity could take advantage of the resources available in their teammates' devices. As a counterpart, these nodes can claim the favor back when needed.

This article presents a study of the collaboration process in a heterogeneous scenario, where handheld devices are combined with PCs in several ways. The interactions among these devices are supported by an overlay network; i.e. a virtual network built over a physical one [4].

This study also explores the advantages and disadvantages, in terms of cooperation level and users' satisfaction, of having a heterogeneous computing scenario. The resources sharing proposal has been evaluated using simulations for several application scenarios. CPU units are considered as shareable resources to simplify the simulations, but in fact they can be any other hardware resource or peripheral; e.g. memory, or time units of a GPS or a Webcam.

Based on the study results we present a set of guidelines for designers of mobile collaborative applications. Integrating these guidelines in the design of the application encourages users to collaborate and share their extra computational resources.

Since software designers need to determine, at an early stage of the development process, the alternative solutions to deal with the devices heterogeneity in a particular work scenario, this article proposes a simulation method based on the study findings. This method can be used not only to determine alternative solutions but also to validate the suitability of already implemented systems when they are used in a specific heterogeneous computing scenario.

Next section describes the problem to address and the research questions of this study. Section 3 presents the related work. Section 4 describes the experimentation setting. Section 5 presents the obtained results and its discussion. Section 6 shows the use of data mining analysis techniques to identify the impact that several variables have on the collaboration process. Section 7 shows the lessons learned. Section 8 describes the simulation method that can be used by software designers to identify or validate possible solutions. Section 9 provides guidelines to deal with resource-sharing issues affecting the collaboration process. Finally, Section 10 presents the conclusions.

2. Problem definition

Various questions arise when we try to analyze the collaboration support provided by several overlay network topologies involving heterogeneous hardware devices; e.g. under which conditions handheld devices are able to share resources with others? What additional resources are needed to encourage collaboration among them?

The definition of a set of relevant research questions will clarify the impact of several network topologies on heterogeneous work scenarios. In particular, it is interesting to study the introduction of powerful devices in a mobile distributed network, which would probably change the network behavior. Therefore the first research question to be addressed is the following one:

Research Question (RQ) 1. *Does a mobile collaboration scenario need extra resources (obtained from other powerful devices) to reach the target cooperation level among the users?*

If the answer is yes, and new resources are required by the network, a new question arises:

RQ 2. *Which effects are caused by the introduction of such powerful devices on the cooperation strategy or behavior of handheld devices?*

Various related works [8, 24, 28, 38] show the network topology characteristics play a role in collaboration, and that it can also encourage and promote cooperation if certain conditions are present. Thus, we not only expect that introducing powerful devices in the network will promote collaboration, but also we believe the collaboration level is highly dependent on network topology and on node distribution. Consequently, a third research question arises:

RQ 3. *Are the topologies used to support collaboration in various fields (e.g. game theory, neural networks, file sharing) applicable to resource sharing on a mobile collaboration scenario?* If answer is yes, then *does the network size affect node behavior?*

Barabasi's studies [5] describe a common behavior pattern, named "networks undergo phase transition", that is found in the nodes of a real or artificially created graph that are "socially related". This phenomenon means that when a given threshold is reached, all the network nodes undergo a transition phase when they start acting as a single entity. Thus, the network properties are shared among all the nodes. This allows us to formulate a fourth research question:

RQ 4. *Do mobile devices follow the phase transition pattern during the resources sharing process?*

Finally, it would be interesting to know if the nodes placement strategy can be used to help improve the collaboration among the network nodes. It gives us a fifth research question:

RQ 5. *Is the nodes placement strategy a variable that can be used to improve the nodes cooperation?*

These research questions are not relevant in some work scenarios like cloud computing, where the collaborative applications have no visibility or control on the network topology. Therefore the nodes cannot take particular actions, e.g. increasing the network degree, to improve the collaboration them.

3. Related work

The collaboration problem among disperse units is present in several areas, such as electrical engineering, electronics and transportation. For example, Ponci et al. [33] propose a context-aware multi-agents framework to dynamically manage energy in complex power systems that involve heterogeneous components and limited resources. Marti et al. [25] present a similar solution to manage the traffic on a road network when there appear meteorological problems. Raveendranathan et al. [36] introduces the concept of virtual sensor, which is composed of body sensors that cooperate to provide specific services in several application domains. Calafate et al. [6] propose a robust and efficient broadcast-based content delivery system for vehicular networks, where the nodes cooperate to minimize the content delivery time. A similar proposal was presented by Kim et al. [20] to minimize the energy consumption during the management of wireless sensor networks.

As previously mentioned, our approach to deal cooperation among the network nodes is based on the use of volunteer and public-resource computing. The volunteer computing concept [40] proposes an interesting idea to share hardware resources among devices belonging to a peer-to-peer network. However, it has well-known limitations generated by the network architecture [9]. Particularly, many nodes (known as free-riders) become selfish and strive to maximize their own utility, by exploiting the system without contributing to the community. Mitigating this effect requires to introduce incentives to encourage the nodes to collaborate with each other.

Some service architectures (e.g. SOA and SODA [10]) and service protocol specifications (e.g. OASIS Web Services Discovery and Web Services Devices Profile [29]) deal with several aspects regulating the interaction among nodes in heterogeneous scenarios. However, such solutions do not impact the collaboration capability of the nodes, e.g. to share resources. Thus, using a solution to enhance the node capability for interaction is as important as using a strategy to promote collaboration among the participating nodes. In that sense this paper addresses just the strategies to enhance collaboration.

A well-known mechanism designed to cope with the free-riding problem is the tit-for-tat repayment, which is a direct reciprocity scheme [28] used by peer-to-peer file-distribution tools, e.g., BitTorrent [35]. In tit-for-tat every node collaborates or not with another one, depending on the last action taken by the latter (i.e. it generates reciprocity actions).

Another approach to deal with the collaboration problem comes from the heuristics used to place resources in a network in order to reach a certain routing objective. This objective may be, e.g., to improve resiliency or performance of a routing protocol [37].

Although these proposals have shown to be successful, they consider all nodes as similar in terms of features and capabilities. Therefore they cannot be directly used to address cooperation in heterogeneous scenarios. Moreover, such proposals consider a network with stable topology, which is not representative of mobile collaboration scenarios.

Several approaches have been used to study the impact of overlay network topologies on the cooperation process; for example to analyze the topology properties in real-world applications with a good cooperation level. In that sense, Iamnitchi et al. have studied the patterns and properties of network topologies in file sharing applications [23], and Lozano et al. have studied them on email systems [24]. The topologies used in our study are based on these last two works.

Another approach to deal with the problem of sharing resources among distributed nodes was proposed by Feldman et al. [11]. They showed the proportional-sharing mechanism achieves a balance of high efficiency and fairness degree at the equilibrium. However, this approach is complex to implement in a decentralized way with only local information.

Nowak studied properties of network topologies for encouraging cooperation and proposed mechanisms for cooperation evolution based on natural selection [28].

Studies by Cassar [8], Santos et al. [38] and Lozano et al. [24] also show the potential impact of the network topology on the cooperation process. Our study basically differs from these previous ones because we model the heterogeneity and limitations of computing devices. This study also takes into account the overlay network characteristics (e.g. the clustering coefficient and the degree distribution) and the placement of devices within the network.

Like these related works, we have limited the study to the ideal environment for the nodes behavior. We then do not take into account other computational effects or drawbacks like load and task balancing or allocation. We do not take the nodes mobility into account either, but we consider changes in the network topology. Simulating mobility of real-world nodes is a complex task [39], which should be addressed once the effect produced by the other network features has been understood. This study used data mining techniques to understand the process of collaboration among nodes and even to predict it.

4. Experimentation setting

All experiments of this study were done with a network simulator, because the tests can be reproduced and also the variables to be studied can be isolated. Each experiment included simulations performed over a discrete scenario with 250 rounds. Previous experiments by Vega [41] have shown this number of rounds is enough to obtain significant statistical conclusions after discarding the first fifty transitional rounds. In the time-based simulations, however, these 50 first rounds are included.

In the experiments, the nodes played a version of the Prisoner's Dilemma game [34] in which each participant plays the game against all neighbors and no one knows the total number of rounds of the game. Each node follows with the same probability a tit-for-tat strategy to decide which action to take.

The simulation process starts by loading a non-weighted topology graph, which is artificially created. It continues with the setting of the variables that represent the environmental conditions. Then, the simulations are executed with only one independent variable, assuring thus that the results reflect the impact of just such variable. Finally, the results are gathered after running a large number of simulations.

The simulations validation is performed using two techniques (1) checking internal simulator invariants and (2) performing preliminary simulations with tested data and known results. This validation is reported in [41]. The preliminary validation tests were made on small network topologies and their results can be verified with mathematically known solutions.

This section describes the network topologies used in experiments, and the variables influencing system behavior. Moreover, the algorithms, metrics and node game strategies used in this study are also described.

4.1. Network topologies

It has been shown there are some topological patterns and graph characteristics promoting cooperation among the nodes in real world networks [23, 24]. The topologies selected for our study include are the most commonly used to promote collaboration (Fig. 1): *torus*, *random*, *power law* and *small-world*. These topologies are briefly explained below.

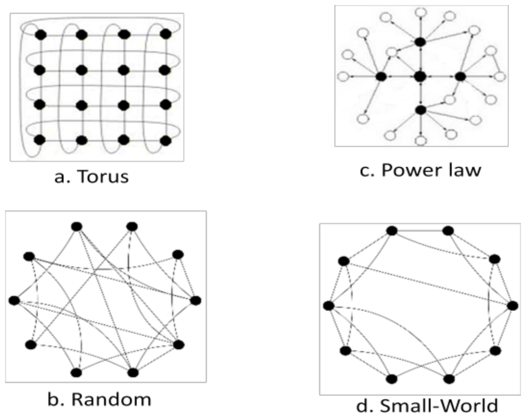

Fig. 1. Network topologies used in the study.

Torus: This topology involves an N-dimensional torus, defined as the Cartesian product of N rings. Symmetric torus topologies are built from rings of the same length. A constraint is that the number of nodes must be D^N, where D is the number of nodes in the ring. Under uniform traffic circumstances, torus topologies have the advantage of providing a balanced use of the network resources. A symmetric torus improves the mesh by connecting the head node with the tail node in each row and column. Thus this topology eliminates the edge effect.

Random: The Waxman's probability model for interconnecting nodes [7, 43] was used to build a random network topology based on Erdős–Rényi model.

It was done using the BRITE topology generator [26]. Edges are introduced between pairs of nodes u, v with a probability that depends on the distance between them. The {u, v} edge probability is:

$$P(\{u,v\}) = a \cdot e^{\left(\frac{-d(u,v)}{b \cdot L}\right)} \quad (1)$$

where $d(u,v)$ is the Euclidean distance from node u to v; a is the probability of edges between any vertex in the graph and controls the average degree of the network, b is the ratio between long and short edges and L is the maximum distance between vertices. Waxman parameters are chosen with default generator values a = 0.15, b = 0.2; and L = 1.000.000 is selected to represent a square surface of side 7.071 points long. As nodes are distributed uniformly random along the surface, $d(u,v)$ is our random variable [0, L]. Thus, the Waxman model generates networks with lower variability of nodes degree and smaller diameter size than other Internet topology generators [43]. The model also has an exponential clustering coefficient distribution, independent of network size which is representative of most random graphs.

Power law: In this topology, a small number of nodes act as hubs (having a high degree), while most nodes have a low degree. Our power law network was created using Barabasi's algorithm and implemented using the BRITE topology generator. According to the incremental growth of the nodes' power degree, the probability of interconnecting a new node u with node v belonging to the network –the {u, v} edge probability– is given by:

$$P(\{u,v\}) = \frac{d_v}{\sum_{k \in V} d_k} \quad (2)$$

where d_v is the current degree of node v to which node u would be attached, V is the set of nodes which joined the network. The lower term is the sum of out-degrees of nodes that previously joined the network.

Small-world: Two characteristics distinguish this topology from other kinds of networks: (1) there is a small average path length between each couple of nodes and (2) there is a high clustering coefficient which is independent of the network size.

Table 1 summarizes the main characteristics of the used network topologies. Their properties allow us to evaluate the effects on the cooperation process. The topological metrics used are the following: *degree distribution*, *average path length scalability*, and *clustering coefficient scalability*.

Table 1

Topology properties

	Degree distribution	Average path length scalability	Clustering coefficient distribution
Torus	Constant	O(N)	Constant
Random	Low variability	O(Log N)	Exponential
Power law	Power law	O(Log log N)	Power law
Small-world	Heavy-tailed	O(Log N)	High variability

– *Degree distribution*. The degree of a node in a graph is the number of arcs connecting to other nodes. The degree distribution is the probability distribution of these degrees on the whole topology.

– *Average path length scalability*. The average path length (APL) is the average number of hops between each pair of graph nodes using their minimum path. The APL scalability shows dependency between the APL and the network size (N).

– *Clustering coefficient scalability*. The local clustering coefficient of a node in a graph is the proportion of vertices of the node to the number of all possible vertices. The clustering coefficient distribution is the probability of these coefficients in the whole network.

4.2. Modeling the network nodes

Each computing device was modeled as being of one of two types: (1) *handhelds*: smartphones or similar, and (2) *PCs*: laptops/desktop PCs or similar. A more detailed classification can be done to get more accurate results, but this classification is enough to understand the main issues affecting cooperation.

These devices were modeled as having only one resource to share and use, namely their CPU slots. The relation between the CPU speed of handhelds and PCs used in the simulations was based on tests done to typical commercial products. We consider a ratio of 1:5 between the processor speeds of these devices. Consequently, we also consider handheld devices are nodes having up to 3 CPU slots to use or share, while PCs have up to 15 slots.

Each resource request has two parameters: (1) the number of CPU slots required by the requester, and (2) the number of rounds for which the requester needs those resources. Both parameters have a maximum allowed value that was defined to assure a minimal quality of service to the nodes.

The maximum number of CPU slots per request was set to 10. Besides, the maximum number of rounds to request a resource was established in 3. This is a simplified model, but more complex task as workflow executions [13] can be modeled in a similar manner. Since each simulation had 250 rounds, which allow us to generate an important number of interactions among nodes and thus we can study the network behavior under several conditions.

The prisoner's dilemma (PD) [34] was the collaboration strategy used to study the relationships between the nodes behavior for the previously presented topologies. In this game two players choose between cooperation or defection. The payoffs for the two actions are shown in Table 2.

Table 2

Payoff matrix of the prisoner's dilemma

Player decision	Co-player Cooperate	Co-player Defection
Cooperate	b - c	C
Defection	b	$\varepsilon \rightarrow 0$

The relations between different possible payoffs follow the rule $b > c > \varepsilon \rightarrow 0$, which immediately poses the dilemma: if cooperation is costly for the individuals and it benefits only the interaction partners, then Darwinian selection should favor non-cooperating defectors and eliminate the cooperators. This leads to a highly inefficient outcome compared to the results obtained by two cooperators.

The decision of choosing a collaboration strategy must be based on the trust in others and their reciprocity. An interesting scenario for the collaboration process is the one where two or more nodes play more than one PD round. In that case, the nodes know the intentions of others and all become only partially anonymous. A common strategy to maximize the payoff is the tit-for-tat strategy. In this strategy and for a given round, a certain node *A* responds to the request of a node *B* making the same decision (i.e. cooperation or defection) made by *B* during the last request from *A*. As a result, tit-for-tat optimization consists in not being harmed, but rather benefited, in comparison with other players.

4.3. Metrics for resources-sharing evaluation

The metrics used in this study were selected mainly taking into account the handheld features, because they are the most limited devices. However these metrics are also applicable to PCs.

One such metric was the *Node Success Percentage (NSP)*. The NSP of a node is defined as the ratio of the satisfied requests to the total of requests made by the node. The average NSP of all the network nodes is the *cooperation coefficient* for that topology.

The second metric we used is the *Node Acceptance Percentage (NAP),* which is defined as the ratio of CPU slots that the node is willing to share to the CPU slots requested by the node. The average NAP for all nodes of the same type in the network represents the *cooperation willingness* of that type of node.

4.4. Algorithm of a collaboration round

We describe below the game algorithm played by the network nodes during a collaboration round. This algorithm includes four stages [41]: *request*, *response*, *evaluation* and *closing*.

Request stage: Each node that itself has no pending tasks creates a new request with a 50% probability of success. First, the node does a self-request for the number of CPU-slots that it needs. However, if its own resources are not enough to complete the task, it sends the number of unitary requests needed to accomplish the task to the neighboring nodes.

Response stage: The strategy to address the response process is simple. Each node having the requested resources responds affirmatively to its own requests. Then, each node that has received a request from other nodes decides whether to share or not its resources following a tit-for-tat policy. Thus, a node v will respond affirmatively to a resource request if the node has free CPU-slots, and also if the requesting node *(u)* is a cooperator. Considering node *u* as cooperator means that such node must have responded affirmatively to the last request from node *v*. If *u* and *v* have not met before and the free CPU-slots condition is accomplished, then node *v* is considered cooperator with 50% of success probability.

Evaluation stage: Each node evaluates its own affirmative responses and assigns CPU-slots accordingly. Then, the nodes evaluate the affirmative responses from other nodes and randomly discard all the excess of offered CPU-slots. Finally, the nodes compute the number of affirmative or negative responses and consider that for the next tit-for-tat decision.

Closing stage: Each node computes pending tasks, taking into account the remaining time for each task, updates its status by removing messages and by calculating statistical data from the previous round.

5. Obtained results

The simulation results allow knowing the range of values that can be found in mobile collaborative scenarios for various topologies and network parameters. These results also give further insights for answering the RQs stated in Section 2. The next subsections describe the settings of the simulations, the obtained results and the conclusions of their analysis.

5.1. The impact of introducing powerful nodes

The simulations show the effect of introducing PCs on a mobile collaborative network, when the nodes play a PD game using a tit-for-tat strategy. Fig. 2 shows the cooperation coefficient (average, maximum and minimum) achieved by handheld devices on the four topologies considered in this study.

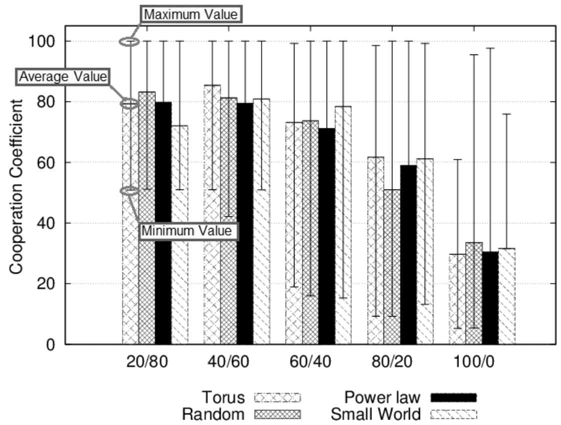

Fig. 2. Cooperation coefficient of handheld devices using four network topologies and five PCs ratios.

The simulation also considers several ratios of handhelds and PC. They are represented in the X axis (Fig. 2) as % of handhelds / % of PCs. For example, the first dataset was obtained with a network of 1000 nodes that had 20% of handhelds and 80% of PCs.

In every simulation (unless otherwise stated), the devices were randomly placed on the network without following any particular topology pattern.

Fig. 2 shows minor variations in the maximum values of the *cooperation coefficient* between networks nodes, when considering various handhelds vs. PCs ratios and also network topologies. The variations are important when the network is composed just by handheld (i.e. a 100/0 ratio). In that case the average values of the cooperation coefficient are considerably lower than in the other network configurations. These results help us to answer **RQ 1**, showing the introduction of PCs improves the level of cooperation among nodes, regardless of the network topology. PCs increase the number of resources available in the network providing thus more collaboration capability to handheld devices.

Although the maximum value obtained for the cooperation coefficient is close to 100% even when there are 20% of PCs, the minimum value decreases with the percentage of powerful devices. Additionally, the effect of increasing the number of PCs improves the minimum value of the cooperation coefficient in the network. We can see that there is almost no improvement in such minimum value (less than 7%) after introducing more than 60% of PCs.

These results show it is possible to improve the cooperation coefficient of the handhelds in a mobile resource-sharing scenario by introducing new resources in the network. We used standard statistical methods to compute the confidence interval and margin of error [31] of the values. The standard error of the mean values is at most 0.055. The relative margin of error for the mean is at most 12.02% for a 95% confidence level. Thus we can consider the average values computed on this test set are valid.

Comparing the same topology with different ratios we have seen that there were statistically significant differences between means determined by one-way analysis of variance, ANOVA ($F(4,366) = 2.396$, $P = 2.49E-60$). Therefore, we think the observations and conclusions based on this test set are also valid.

5.2. The impact of introducing powerful nodes and the tit-for-tat game strategy

Although the two types of nodes considered in this study play the same game and follow the same strategy, there is no guarantee that they will have the same behavior or obtain the same cooperation values.

Fig. 3 represents the average willingness of a node to cooperate with other nodes of its community. This simulation included 1000 nodes arranged according to the four chosen topologies. The network was composed of 60% handhelds and 40% PCs, and it had an average degree equal to 6.

Moreover, we also ran the same experiment but with the cooperation willingness calculated taking into account the tit-for-tat strategy and unlimited available CPU slots for the participating nodes. The "limited" and "unlimited" tags, respectively, differentiate these simulations. The difference in the results shows that the cooperation willingness from the nodes does not correspond to their real willingness, due it is influenced by their resource limitations.

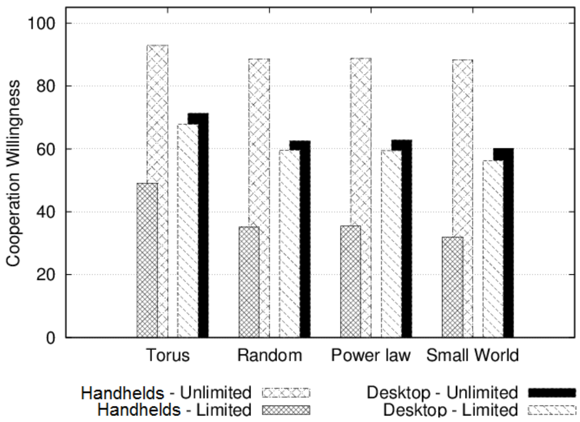

Fig. 3. Cooperation willingness of powerful and handheld devices with limit/unlimited neighbor resources.

These results help us to answer **RQ 2**, indicating that the cooperation willingness decreases with decreasing availability of CPU slots. Thus, this limitation affects mainly handheld devices.

Since PCs have many more CPU slots available than handhelds, the difference in the willingness of PCs to cooperate is lower (between 2.96% and 3.9%) than handhelds, independently if we consider limited or unlimited CPU slots. Typically, PCs are self-served although the resources are scarce for handhelds. Therefore such a situation considerably degrades the cooperation coefficient of the network: for non-torus topologies it is more than 50%.

These results show that introducing powerful devices to the network and adding some limitations that did not originally exist in the PD game, have a considerable impact on simulation results. In spite of having several handhelds willing to cooperate, many of them finally do not share their resources due to the limited number of available CPU slots.

The series in Fig. 3 correspond to the sample mean of the test sets. The standard error of the mean values is at most 0.005. The relative margin of error for the mean is at most 1.61% for a 95% confidence level. Therefore, we can consider these values as valid.

Comparing the cooperation willingness values for limited and unlimited handheld devices, we can see that there are statistically significant differences between means determined by t-test (unpaired samples and unequal variances), when $\rho = 1.33E\text{-}26 < 0.05$.

If we compare the cooperation willingness results on unlimited handheld devices for the four topologies, we also can see statistically significant differences between the means determined by a one-way analysis of variance, ANOVA ($F(3,996) = 2.614$, $P = 6.45E\text{-}25$). Thus, these observations can be taken as valid.

5.3. Impact of using several network topologies

This simulation scenario is the same than the previous one, but in this case the observed variable was the *Cumulative Distribution Function* (CDF) of the requests failure percentage of handheld devices. This CDF value represents the percentage of nodes that keep a certain defection level when trying to collaborate with the neighbors. The X axis in Fig. 4 represents the percentage of requests that failed because the potential collaborator had no available CPU slots for sharing. The curves shown in Fig. 4 indicate for a given percentage of failure, the percentage of requesting nodes having these failures, e.g., in a network with a torus topology, 60% of the failures occur to just 5% of the requesting nodes.

The simulation results are analyzed below in three different areas: (1) below 40% of failure percentage, (2) from 40% to 80% and (3) above 80%. In the first area, there is a linear increase of the network nodes affected by the failed requests; i.e. up to 40% failure in the requests, the number of affected nodes grows almost linearly. This means all topologies, but particularly random, are well-influenced by topological properties and potentially the network can achieve better *node success percentage* (NSP).

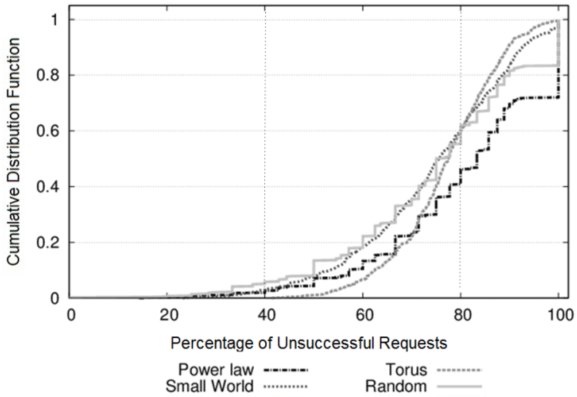

Fig. 4. CDF of the failure percentage of handheld devices.

In the second area, the torus CDF grows faster than in other topologies. In the case of the power law topology, the CDF does not grow as much as in other topologies, which indicates that there are many underused resources. This resources underuse clearly negatively affects the cooperation rate among nodes.

Finally, the third area analyzes the random and small world networks. The first one has almost a 20% of their handhelds with an NSP lower than in the small-world, even if the neighbors of the requester node have CPU slots available. In small world, the distribution is similar than a torus network.

Fig. 4 also shows a local point of view of cooperation among nodes and it allows us to decide among various topologies without considering the effect of the game played by the nodes and the cooperation strategy used by them. Based on these results, the recommendation is to consider the small-world topology as the best one. Its minimum failure percentage is not as good as random topology –and not as bad as power law–, but its distribution is somehow similar to that of the torus topology. These results answer **RQ 3** indicating the studied network topologies produce different impacts on the collaboration; thus some topologies are more suitable than others.

5.4. A network undergoing phase transition

This simulation intends to identify whether Barabasi's phenomenon [5] can be found in heterogeneous networks during resources sharing (**RQ 4**) or not. In order to do that we ran a complete set of simulations, which involved four network sizes (5000, 1000, 500 and 100 nodes) and the four selected network topologies. The ratio of handheld was kept at 60% and the nodes degree was kept at 6. The observed variable was the node success percentage (NSP). After running 250 rounds we computed, for each node in the network, the number of rounds in which such a node made a collaboration request, divided by the percentage of success obtained by those requests (Fig. 5).

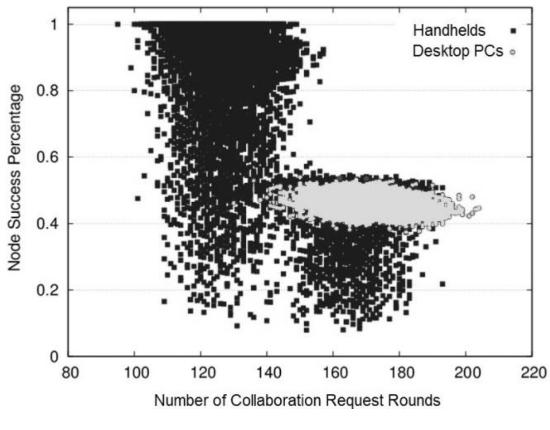

Fig. 5. Node success percentage vs. number of collaboration request rounds.

The results indicate that the "undergo phase transition" phenomenon can be found in heterogeneous networks, where almost all the nodes have cooperation coefficient around two values: 50% or 100%.

It is interesting to note that the PCs behave as expected; i.e. they introduce new resources to the network, thus helping handhelds but without being harmed by the system and getting around 50% of cooperation in all cases. Instead, the handhelds cooperation is distributed from 12.5% to 100% of NSP.

The results shown in Fig. 5 imply that handhelds can be divided into two groups: "aggressive" and "moderate". When handhelds act aggressively, they perform a higher number of requests –equivalent to PCs in number– and their cooperation coefficient falls below 50% –the same achieved by PCs–. This is caused in part because handhelds do not have available resources equivalent to their requests, and thus their requests frequently fail.

Moderate handheld devices –those that make fewer requests than PCs –are not harmed because the number of requested CPU slots is equivalent to the number they offer to others. These two behaviors of handhelds generate two different NSP distributions (Fig. 5) and they cause the presence of a "network undergoes phase transition" phenomenon.

Fig. 5 also shows that a relevant collaboration process starts just when at least 50% of the requests are positively answered. Then, the percentage of nodes involved in a collaboration process remains stable until the cooperation coefficient goes over 85%.

5.5. Impact produced by the network size

This section shows the results of evaluating characteristics related to network size, which allows us to know whether or not this feature produces changes in the system behavior. It also helps us to address the second part of **RQ 3**.

The impact of network size is related to two network parameters: the nodes' degree and the number of nodes in the network. We have kept the average degree constant and have modified the number of nodes, which allows us to study network size impact.

Like the previous tests case, this simulation involved random networks with four sizes: 5000, 1000, 500 and 100 nodes. The network degree remained at 15 and the percentage of handheld was 60%.

It seems reasonable to argue that if a node only interacts with its neighbors and if the network size increases without changing the average nodes degree, then the node behavior should not change. However, the results shown in Fig. 6 demonstrate that there exists a propagation effect between nodes that modifies their behavior linearly with the network size. Small networks assure a high minimum cooperation level among handhelds, but these networks require that an important number of nodes be willing to collaborate.

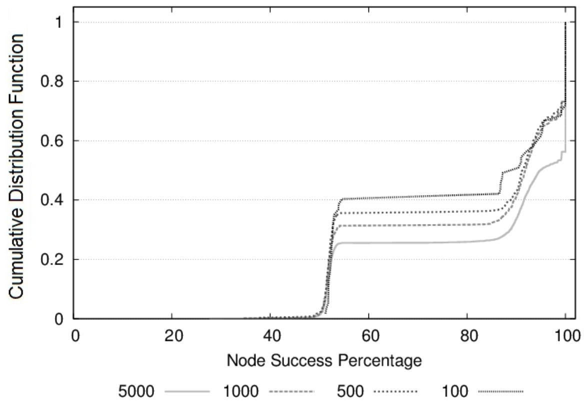

Fig. 6. CDF vs. Node success percentage of handheld devices.

From the network size point of view, the decisions of each node have a different impact on the global results. The larger the network is, the more willing will be the node to share its resources. Thus, the global cooperation willingness improves as expected.

Fig. 6 also shows three different patterns of behavior for CDF values. This behavior, particularly the stationary values, can be explained by the *networks undergo phase transition* phenomenon, explained in Section 2. It supports the results shown in section 5.4 which helped us to answer **RQ 4**.

Fig. 7 shows the network size effect from the point of view of nodes behavior. It depicts the histogram of CPU slots dedication on both types of devices on a random topology with 60% of handheld devices and two different network sizes: 125 and 1000 nodes. The average network degree was kept at 6.

On the 125 nodes network, both types of devices have a high percentage of free CPU slots. This effect is justifiable because the tit-for-tat game strategy gets the fairest price for the shared resources. However, it is a clear example of underused resources on the system. Moreover, when we increase the network size, both systems reduce the number of free CPU slots in order to share more. A notable consequence is that handheld devices become fairer as the difference between shared and used resources becomes smaller.

Another effect is that PCs brake their balance between shared and used CPU slots, in favor of increasing the resources for sharing, which reduces the number of requests. Therefore, the PCs not only keep the cooperation coefficient at 50% (Fig. 5), but also they also are more willing to help handheld devices.

These observations allow us to answer the second part of **RQ 3** indicating that network size clearly affects the collaboration strategy used by handhelds, and thus it affects the whole system. The next subsection analyzes the potential relationship between nodes degree and node success percentage (**RQ 6**).

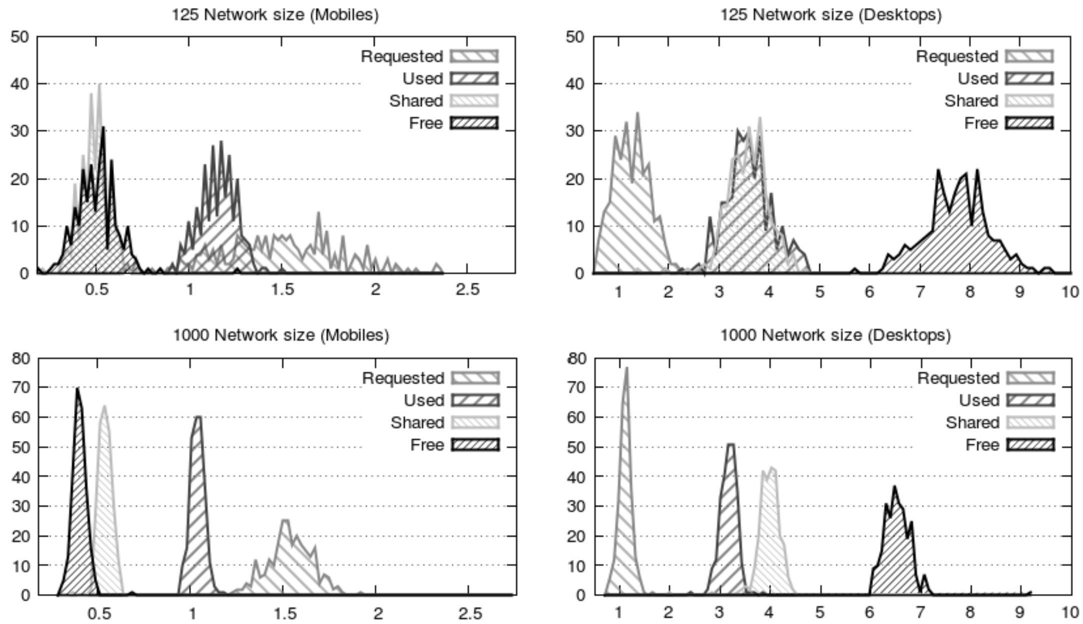

Fig. 7. Histograms of average CPU slots requested, used, shared and free on random topologies.

5.6. Impact produced by the nodes degree

A simulation was performed to identify the impact of nodes degree on the success percentage of collaboration requests. It was done with a network of 5000 nodes, with 60% of PCs, and random, power law and small-world topologies. Two average values for the network degree were used: 15 and 35. The observed variable was the CDF of the handheld nodes success percentage. Fig. 8 shows that the cooperation coefficient is inversely proportional to the network degree, which allows us to answer **RQ 5** and **RQ 6**.

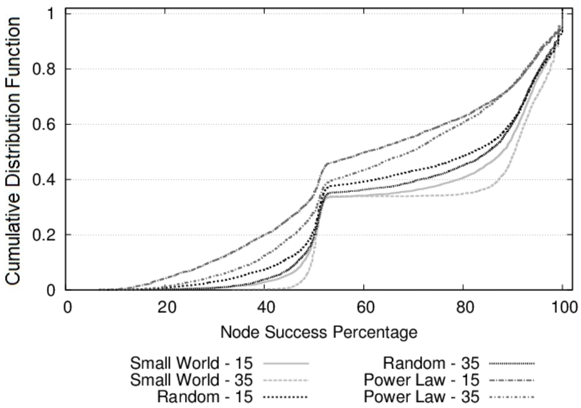

Fig. 8. CDF vs. node success percentage of handheld devices.

This result reflects the fact that tit-for-tat positive decisions are fast in being flooded through the network. Consequently, we can affirm that the nodes connectivity also contributes to trigger a “networks undergo phase transition”. Finally, it is interesting to point out that the phase transition effect cannot be generalized to all topologies because, at least, it is dependent on the network degree.

6. Analysis of results using data mining techniques

Features selection is an important step when datasets have many variables and correlated data. For an efficient identification of structures with data mining techniques, it is needed to evaluate and determine the variables in the data set that contain valuable information [19]. Determining subsets of a reduced number of features, e.g. with machine learning techniques, is another way to identify efficiently properties in data sets [2]. The feature subset selection in particular is a method for enhancing the performance of data mining algorithms by reducing the variables search space [14]. With this analysis we intend to: (1) identify a short list of features to understand resource-sharing processes in a mobile scenario, and (2) evaluate each feature algorithmically using some well-known feature ranking algorithms.

We create our own datasets for this analysis. The datasets are composed of all requests, responses, NSP and states of each node, round by round. As was previously mentioned, each simulation included 250 rounds. Over 100 simulations were run by combining the features previously explained; i.e. the four selected topologies, with the five ratios of handhelds/PCs and to the four network sizes.

6.1. Features set

Choosing an appropriate feature set is the most critical part of any machine learning algorithm. We have followed the steps and recommendations presented in [14] to choose the appropriate feature set and analyze it. Although there are techniques and algorithms to construct this feature set, we have created a set of "ad hoc" features because the data domain is already known to us.

Understanding the resource-sharing process requires identifying features that represent the system components, particularly the devices, topologies and node proximity. The feature dataset describing these system components is typically obtained by aggregating their individual features. An aggregation function is required to combine the feature values for obtaining a whole dataset that is going to be analyzed. We have named *device features* the dataset that represents the network nodes. The relationship among nodes has been captured in a dataset called *proximity features*. Finally, the network topology was represented through the *topological features* dataset. The features selected to create each of the mentioned datasets are presented below.

Device features:

- *CPUs*. This feature indicates the available CPU slots (i.e. resources) that the device has for using and sharing. This feature is intended to embed the fact that a device with more CPU slots is more likely to share resources.
- *Device type*. This feature represents the type of device; i.e. handheld and PC. It is similar to the *CPU* feature, but instead of being a numeric value it has a nominal one.
- *Request rounds*. It indicates the number of rounds in which a device has requested CPU slots to other nodes. Since this feature is the denominator of the NSP metric, it seems reasonable that the NSP val-

ue of a node depends on the number of *request rounds* in which that node has participated.

- *Needed CPUs*. This feature is the number of extra CPU slots (i.e. CPU slots from neighboring nodes) required for a node to perform a specific task.

Topological features:

- *Topology*. This feature represents the type of network topology: torus, random, power law and small-world. The simulation results have shown that the devices behavior (i.e. the collaboration coefficient) depends on the network topology.
- *Network size*. It indicates the number of nodes composing the network. The results shown in Section 5 indicate that the devices behavior also depends on the network size.
- *Clustering coefficient*. The clustering index measures the local density of nodes in a network. Other researchers have reported that nodes in a dense area are more likely to share resources compared to those located in a sparse area [1, 24].
- *Degree*. This feature is the connectivity of a node and it is intended to measure the fact that a more connected node is more likely to share resources.
- *Neighbors' degree.* This feature represents the aggregated degree values of the neighbors of a certain node. While the connectivity of a node is significant, the connectivity of its neighbors sometimes also plays an important role. When nodes have many neighbors, it is possible that those latter nodes are more likely to cooperate.

Proximity features:

- *PCs*. This feature counts the number of powerful devices in the neighborhood. The PCs have more CPU slots than handhelds. This difference allows them to share these resources.
- *Neighbors' CPUs*. This feature is the aggregated number of CPU slots in the neighborhood of a node. This feature is related to the previously presented *PCs* feature.
- *Two-hops neighbors' CPUs.* This feature is related to *neighbors' degree,* but it aggregates the CPU slots for an area of two hops for a particular node.

Note there are couples of features that have high correlation (i.e. similarity) between them. In that case it is enough for this analysis to use one of them. For example we have not used the *device type* feature because it provides information similar to *CPUs*. In this case, we have chosen *CPUs* because it has a numeric value for the analysis. Likewise we use *Request rounds* instead of the *Needed CPUs* feature.

6.2. Features selection algorithms

The objective of these simulations is to execute the feature selection algorithms, which will suggest us a subset of relevant features to analyze during the study of the cooperation coefficient in several scenarios. The features selection experiments were conducted using filter-based approaches on the datasets.

Two well-known feature selection algorithms were applied: (1) *Correlation-based Features Subset Selection* (CFS), an algorithm that evaluates the feature subsets, and (2) *ReliefF*, an algorithm that evaluates individual features. We have chosen these algorithms because they provide reliable feature sets, they are able to process continuous variables, e.g. the cooperation coefficient, and they let us understand the resource-sharing processes. Other algorithms can provide similar or even better results, but the underlying process is more complex to understand [19].

The Weka machine learning framework [15] provided the implementation of these algorithms. We briefly review these algorithms below.

Correlation-based Features Subset Selection (CFS) [16]: CFS evaluates the usefulness of a subset of features by considering the individual predictive capability of each feature, along with their degree of redundancy. The algorithm selects subsets of features that are highly correlated with the class, but having low correlation between them.

ReliefF [22]: This algorithm evaluates the usefulness of a feature by repeatedly sampling an instance and considering the value of the given feature for the nearest instance of the same and different classes. We have chosen it because it is noise-tolerant and unaffected by feature interaction. However, ReliefF searches for all the relevant features, even if they are redundant. This algorithm assigns a "relevance" weight to each feature. In the simulations, we selected features with a relevance ranking above 0.

6.3. Selecting features for analysis

The algorithms must rank the features according to their relevance values or order of importance. The *topology* feature is highly relevant in this study, since the four selected topologies do not have the same influence on the cooperation coefficient. In our preliminary tests, we have seen that the selected features and their order depend on the *topology* feature.

We classified the datasets into four groups to show the relevance of the topology. Table 3 shows the results (i.e. the selected and ordered features) after applying the algorithms to all datasets.

We have used ten-fold cross validation in the evaluation of the features selection process [17]. For the CFS algorithm, we reported the number of times that a feature was selected in the ten-fold cross validation. For ReliefF algorithm, we reported the average relevance of the ten relevancies from the cross validation (selected features had a relevance ranking above 0).

Both algorithms selected the *Request rounds* feature as relevant for almost every dataset. Notice the CFS algorithm does not select features with high correlation; e.g., it selects only the most relevant one between *PCs* and *Neighbor CPUs*. Therefore the set of features selected by CFS tends to be minimal. However the ReliefF algorithm selects all relevant features, even if there is similarity among them.

A first analysis of these results indicates that the cooperation coefficient in the *Torus* topology depends mainly on the *proximity* features (i.e. *Neighbor CPUs, Two-hops neighbor CPUs* and *PCs*). This is an expected result because the *Torus* has no topological properties differentiating its nodes.

The cooperation coefficient in the *Random* topology depends on the *proximity* features (i.e. *PCs* is the most influential feature), due to the low variation in the topological properties of the nodes. In the case of the *Power Law* and the *Small-World* topologies, the nodes behavior depends on *network topological* features. In particular, the cooperation coefficient depends on the *Clustering coefficient*.

The set of relevant features obtained by both algorithms seems to be similar, but it cannot be clearly seen by just analyzing Table 3. In order to check such hypothesis, we re-processed the results obtained by the ReliefF algorithm. This algorithm identifies the similarity between the selected features and thus helps to reduce the feature set. As mentioned above, we know the result of CFS tends to be minimal, and therefore it does not need to be re-processed.

To reduce the ReliefF resulting feature set while keeping the optimal salient characteristics of the data, we applied Principal Component Analysis (PCA) transformations [17]. PCA is a procedure using an orthogonal transformation to convert a set of possibly correlated variables into a set of uncorrelated variables called *principal components*. Dimensionality reduction is achieved by keeping the components with highest variance. The number of principal components is less than or equal to the number of original features.

Table 3

Selected and ranked features by topology

Torus	Random	Power Law	Small World
CFS			
- Request rounds (10/10) - NeighborCPUs (10/10) - Two-hops neighbor CPUs (10/10)	- Request rounds (10/10) - PCs (10/10)	- Request rounds (10/10) - PCs (10/10) - Clust. coef. (10/10)	- Request rounds (10/10) - PCs (10/10) - Clust. coef. (10/10) - CPUs (10/10)
ReliefF			
- Neighbor CPUs (0.004) - Two-hops neighborCPUs (0.002) - Network size (0.001) - PCs (0.001) - Request rounds (0.001)	- PCs (0.009) - Degree (0.009) - Neighbor CPUs (0.007) - Clust. coef. (0.002) - Request rounds (0.001) - CPUs (0.001) - Network size (0.001)	- Degree (0.007) - CPUs (0.002) - PCs (0.001) - Clust. coef. (0.001) - Neighbor CPUs (0.001)	- PCs (0.017) - Neighbor CPUs (0.015) - Clust. coef. (0.004) - Two-hops neighbor CPUs (0.003) - Degree (0.002) - Network size (0.002)

Applying a PCA transformation can reduce the eight originally selected features (i.e. *CPU*s, *Degree*, *Clustering coefficient*, *PCs*, *Request rounds*, *Two-hops neighbor CPUs*, *Network size* and *Topology*) to only 6 truly independent ones. Dimensionality reduction is accomplished by choosing enough eigenvectors to account for some percentage of the variance in the original data (default 95%). Attribute noise can be filtered by transforming to the principal components space, eliminating some of the worst eigenvectors. After applying the PCA transformations the obtained results indicate that the relevant features selected by both algorithms are similar, and they correspond to mainly those identified by CFS.

6.4. Using selected features by examples

The results obtained from feature selection analysis allowed us to observe two clear implications: (1) the network topology is relevant to improve the cooperation coefficient and (2) there are different features to define "well-placed" handheld devices in each type of topology.

Another important issue addressed in this study was the distribution of the various types of devices over the network in order to maximize the cooperation possibilities (**RQ 5**). In order to answer this challenge, we ran several simulations to evaluate the sensitivity of the cooperation coefficient to different nodes' distributions, such as node degree and clustering coefficient. With this set of experiments we tried to achieve an optimal network operation, but the results showed that there is no significant difference among several nodes distributions (Table 4).

Table 4

Cooperation coefficient of handheld devices for various device placement strategies

Topology	Handhelds placed on	Min.	Max.	Average
Power law	Clustering	39.89%	100.00%	79.93%
	Lower degree	40.76%	100.00%	74.25%
	Higher degree	37.79%	100.00%	80.91%
	Random	26.99%	100.00%	78.55%
Small-world	Higher cluster.	49.40%	100.00%	81.67%
	Lower degree	48.48%	100.00%	81.12%
	Higher degree	50.00%	100.00%	81.42%
	Random	47.78%	100.00%	80.77%
	Higher degree	39.11%	100.00%	80.40%
Random	Random	34.59%	100.00%	81.69%
	Clustering	39.89%	100.00%	79.93%

Since the cooperation coefficient is not sensitive to the analyzed parameters, it is possible to obtain a very robust system with some placement strategies. Therefore a random device placement strategy can give an acceptable Cooperation coefficient.

In order to increase the cooperation possibilities we ran several simulations to evaluate the sensitivity of the Cooperation coefficient to the selected features. From the feature selection results, we chose *Request rounds* and *PCs* for the four network topologies, and we also considered four devices placement strategies. The simulations had 1000 nodes, a degree of 6 and 40% of PCs. These PCs played a tit-for-tat strategy. Fig. 9 presents the computed handhelds' *Cooperation coefficient* of a *Small-world* network, considering the four device placement strategies. Next we explain each of the placement strategies.

a) Random: This is the same strategy used in the simulations shown in Sect. 5. We will consider this strategy as the original one, and therefore we will compare the rest of the placement strategies with it.

b) Requests: In this case the number of handheld requests has been reduced by 20% and the requests of PCs have been increased by 30%. However the total number of requests has been kept invariable; i.e. as in the original simulation.

c) Heterogeneous links: This strategy considers placing nodes in a way that handhelds increase the number of links with the PCs.

d) Requests and heterogeneous links: This strategy mixes the two previously presented ones.

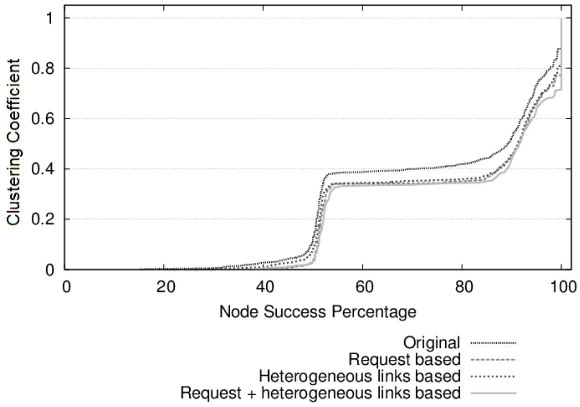

Fig. 9. Node success percentage on handheld devices vs. their clustering coefficient.

Fig. 9 shows that the three new strategies improve cooperation among handhelds, with respect to the original one (random). We can also observe that applying strategy d) does not let the node cumulate the collaboration improvement produced by these strategies separately. The results also show the reduction in the number of requests does not affect the "phase transition effect". These results allow answering **RQ 5**, indicating there exist heuristics for nodes placement that allow improving the node placement cooperation.

7. Lessons learned

A number of issues related to the potential capabilities and limitations of mobile collaborative applications are understandable based on the simulation results. Software designers must consider these issues to ensure the collaborative applications will be suitable in a particular work scenario and also to support a specific mobile activity. The obtained results indicate that the cooperation coefficient decreases when:

– The overlay collaborative network is only composed of handheld devices (Fig. 2). It typically affects the design of solutions supporting collaborative activities with high users' mobility.

– The number of nodes in the overlay network decreases (Fig. 6). In the real world, mobile collaborative networks are formed on-demand and usually they involve a small number of nodes. In that case the users must be physically close during the collaboration process to increase the nodes degree and thus ease the cooperation among devices. Otherwise the collaboration process will be almost impossible to be done successfully.
– Handheld devices are "aggressive" in the sense that they perform a high number of requests (Fig. 5). Therefore, when the available resources are scarce, the collaborative applications must wait for a random time period before sending a new collaboration request. It addresses two situations: (1) to reduce the probability of collision of those requests and (2) to ensure an important rate of the shared resources. These issues impact directly on the response time perceived by the end-user during the collaboration process.
– The number of powerful devices in neighborhood decreases. The software designer, however, can use the handheld device placement strategies to try reducing this impact (Sect. 6). For example, the application could try using "open" mobile devices in the neighborhood, even if they are not involved in the collaboration process, as a way to improve the resources available for collaboration or to manage the current network size or topology.

Furthermore, the simulations allow us to understand the devices behavior from the local point of view and make some preliminary conclusions. For example, tit-for-tat strategies highly harm participants with a larger amount of resources (like PCs) compared to other participants (Fig. 7). This repayment strategy is able to make the game as fair as possible. However, this is not a good global strategy to apply since it generates situations of under-used resources (Fig. 7). Software designers must consider to use different (at least two) repayment strategies in the application, one for handhelds and another for powerful devices. Typically it means to deliver two versions of the collaborative application, which must be able to interoperate between each other.

8. Simulation method

Software designers need to identify, at early stages of the development process, which computing scenario is feasible to use when trying to address collaboration among heterogeneous devices. An early clarification of this issue can help developers to avoid several problems, e.g. writing source code that will not be part of the final system, developing a system that will have performance/scalability limitations, or experimenting a lack of collaboration among devices because there are insufficient resources to perform a certain task.

Based on the study results reported in this article, we have systematized and simplified the performed simulation process in order to make it feasible to be used by software designers during the early stages of a development process. Figure 10 summarizes the resulting simulation method.

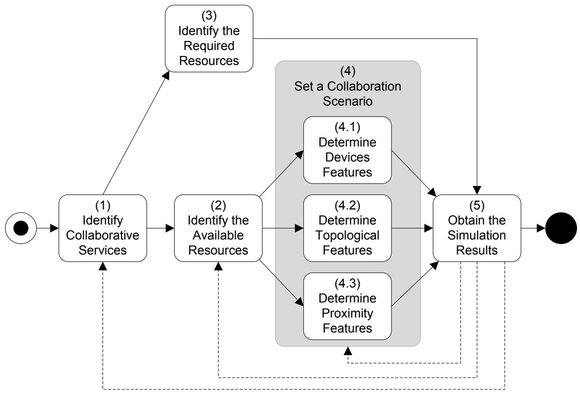

Fig. 10. Structure of the simulation method.

The first step of the process involves determining the set of services that will consume shared resources through the network. Based on that it is possible to estimate, for each type of device participating in the computing environment, the resources that they will have available for sharing with other nodes (step 2) and also those that are required to process the collaborative services (step 3).

The results of step 2 represent the constraints on the solution and the results of step 3 represent the requirements. Considering these two components it is possible to explore several collaboration scenarios to determine which solution can provide a suitable option to deal with the collaborative services (step 4).

Based on the feature set identified and validated in section 6, the features of three main components should be set: the devices (4.1), the network topology to be used (4.2), and the nodes proximity (4.3). Using these settings and the resource requirements identified in step 3, it is possible to automatically obtain various collaboration indicators (e.g. the cooperation coefficient, the cooperation willingness and the CDF) for a heterogeneous computing environment (step 5). Software designers can interpret these indicators in order to determine suitable settings to deal with the collaborative services.

Depending on the indicator values, the designers can perform changes to the feature set, in the resources availability or come-back to the first step to perform a redefinition of the simulated computing scenario. If the indicator values for a certain setting are favorable, then the designers will know that such configuration can be assumed as a structural solution during the next steps of the system development process; i.e. during architectural and detailed design and also during the implementation process.

This simulation method can also be used to evaluate capabilities and limitations of already implemented systems, in term of collaboration for sharing resources. It is possible to obtain the collaboration indicators determining how suitable a collaboration strategy is for sharing resources in a computing scenario by replicating the work environment of an application and the implemented strategy. Therefore this simulation method can be used for both, supporting the design of new solutions as well as evaluating the suitability of already implemented applications.

This simulation process can be implemented through a tool that requires the user select just the simulation options. It will ease the use of the method by software designers.

9. Guidelines to deal with resource-sharing issues

Guidelines that help software developers to deal with resource-sharing in mobile collaboration scenarios are presented below. We show some solutions to implement such guidelines on mobile applications.

9.1. Dealing with network topology

If the overlay network depends on the developers, then their efforts should focus on ensuring that the applications use a small world topology. Kleinberg identified the problem of how to find shortest paths in a decentralized way, in the case of small world network with only local information [21]. A new method of constructing a small world topology in a wireless network was proposed in [18]. Wang and Nakao [42] propose a scheme of evolutionary game theory for topology evolution to change any given overlay topology into the small world structure. If the application runs on a non-controllable topology, software designers should try the network nodes have a large number of links among them.

9.2. Devices placement

Roy et al. [37] proved that the overlay placement problem is a non-deterministic and polynomial-time hard problem. In section 6.4 we have seen that the device placement can be used to improve the nodes cooperation level. Since we use an overlay network, software designers can embed in the collaborative application a software component that is able to take advantage of the existence of a physical and a logical (overlay) network in the collaboration scenario. Such a component should listen to the physical layer of the network and find candidate nodes to be "placed" in the overlay network. Although such devices do not need to be aware of their participation in the collaboration process, the effect produced on the whole system would be similar to place a regular node in the overlay network. Thus, the collaboration capability of the whole system could be improved.

9.3. Network scale

An unexpected result of our study was the fact that the network size affects the cooperation coefficient. Therefore, we recommend that the devices interested in sharing resources should form a single network, because partitioning the network will not increase the cooperation coefficient.

9.4. Handheld versus PC requests

Considering the presented scenarios, if all handheld requests depend on the developers decisions, then the cooperation algorithm should focus on ensuring that handhelds perform fewer requests than PCs; and also that such requests are equivalent to the available shared resources of each node. It has been shown in Fig. 5 that non-aggressive nodes – those making a number of requests equivalent to their own available resources - get a good collaboration rate. By contrast, those nodes having an aggressive behavior frequently fail their requests because we are using a tit-for-tat collaboration strategy, which assures a fair payoff between shared and gained CPU slots. It means that a node should not run a process asking most of the required resources to other nodes, because in that case the unavailability of shared resources will make the system enter in a deadlock.

Furthermore, if the collaboration strategy depends on the developers and powerful machines will be introduced in the network, it is mandatory to consider a different approach (for handhelds) to share resources and increase their usage. The simulation method presented in the previous section can help designers to address these cases.

10. Conclusions and further work

There exists today a many stationary and mobile computing devices that could be collaboratively used to support several activities in areas such as mining, construction, transportation, or manufacturing. In fact, most loosely-coupled activities can be addressed using just mobile devices or embedding a portion of stationary devices. Managing device heterogeneity is an open issue in mobile collaborative systems; therefore there are no guidelines on how to deal with it.

This paper presented a study that tries to understand the challenges behind the process of hardware resource-sharing in mobile collaborative scenarios. The study was based on simulations, which analyzed the impact of several overlay network topologies (particularly torus, random, power law and small-word) on the resource sharing process. The simulations considered two types of network nodes, handheld devices and PCs.

The obtained results allow us to answer six research questions. Some of them focus on measuring the impact of a network feature (e.g. size, topology or degree) on the collaboration process, while others intend to understand the consequence of performing particular actions on the network (e.g. introduce powerful devices or manage the nodes placement) as an strategy to help improve the collaboration rate among participants.

After performing the simulations we did an additional analysis of the network features participating in the system to determine their roles and relative importance in the collaboration process. Known data mining techniques were used in this analysis. The results show the availability of shared resources in local/one-hop/two-hops devices considerably affects collaboration. The network topology, its size, level of clustering and degree also affect such process.

These results provide reusable knowledge for developers of mobile collaborative applications. In order to capture such knowledge and put it available for software designers, a simulation method was defined by systematizing and simplifying the process conducted in the reported study. This simulation method can help designers to both, (1) identify appropriate strategies for sharing resources depending on the computing scenario, and (2) determine the suitability of already implemented solutions to share resources in a certain work environment. The use of this simulation method during the early stages of an application development can help software engineers to avoid several problems.

A set of lessons learned and guidelines have also been included in this paper for helping engineers to design mechanisms that allow mobile users to share their free computational resources almost without affecting the performance of the local applications.

As mentioned above, in this area there are still several open issues and research opportunities. For example it would be interesting to extend the results shown in Sect. 5 to provide a distributed self-placement algorithm based on empirical observations. We believe it is possible to find mechanisms that dynamically include nodes so that the network topology is favorably changed. Based on that solution it would be possible to build self-regulated ad-hoc networks encouraging nodes cooperation.

Another point to address as part of future work is the heterogeneity of the communication links composing the network, which clearly affects the nodes' collaboration capability. In this article we have assumed the transmission capability of the network links is stable (i.e. their quality does not affect the collaboration results), since the simulations try to isolate the effects produced by other variables namely the network topology, the devices population or the network degree. Analyzing the influence produced by heterogeneous communication links on the collaboration process can be the goal of a next study.

Acknowledgements

This work was partially supported by Fondecyt (Chile) grant 1120207, LACCIR grant No. R1210LAC002 and Spanish MEC DELFIN TIN2010-20140-C03-01, and also the Community Networks Testbed for the Future Internet (CONFINE) Large-scale Integrating Project: FP7-288535.

References

[1] M. Al Hasan, V. Chaoji, S. Salem and M. Zaki, Link prediction using supervised learning, in: *Proc. of SDM'06 Workshop on Link Analysis, Counterterrorism and Security*, 2006.

[2] H. Al-Mubaid and D. Moazzam, A Model for mining material properties for radiation shielding. *Integrated Computer-Aided Engineering* 19(2) (2012), 151-164.
[3] D.P. Anderson, BOINC: A system for public-resource computing and storage, in: *Proc. of Fifth IEEE/ACM Int. Workshop on Grid Computing (GRID'04)*, 2004, pp. 4-10.
[4] D.P. Andersen, H. Balakrishnan, M. Kaashoek and R. Morris, Resilient overlay networks, in: *Proc. of the 8th ACM Symp. on Operating Systems Principles (SOSP'01)*, Canada, 2001.
[5] A. Barabasi, Linked: the new science of networks, *Perseus Pub.*, New York, NY. 2002.
[6] C.T. Calafate, G. Fortino, S. Fritsch, J. Monteiro, J-C. Cano and P. Manzoni, An efficient and robust content delivery solution for IEEE 802.11p vehicular environments, *J. of Network and Computer Applications* 35(2), (2012), 753-762.
[7] K.L. Calvert, E.W. Zegura and M. Doar, Modeling Internet Topology. *IEEE Comm. Magazine*, 35(6) (1997), 160–163.
[8] Cassar, A. Coordination and cooperation in local, random and small world networks: experimental evidence. *Games and Economic Behavior* 58 (2007), 209-230.
[9] C. Cusack, C. Martens and P. Mutreja, Volunteer computing using casual games, in: *Proc. of Int. Conf. on the Future of Game Design and Technology*, London, Canada, 2006.
[10] S. de Deugd, R. Carroll, K.E. Kelly, B. Millett and J. Ricker, SODA: Service Oriented Device Architecture, *Pervasive Computing* 5(3) (2006), 94-96.
[11] M. Feldman, K. Lai and L. Zhang, The proportional-share allocation market for computational resources, *IEEE Trans. on Parallel and Distributed Systems* 20(2009), 1075-1088.
[12] L. Guerrero, S.F. Ochoa, J. Pino and C. Collazos, Selecting devices to support mobile collaboration, *Group Decision and Negotiation* 15(3) (2006), 243-271.
[13] J.O. Gutierrez-Garcia and K.M. Sim, Agent-based Cloud Workflow Execution, *Integrated Computer-Aided Engineering* 19(1) (2012), 39-56.
[14] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *J. of Machine Learning Research* 3 (2003), 1157-1182.
[15] M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann and I. Witten, The weka data mining software: an update, *ACM SIGKDD Explorations Newsletter* 11(1) (2009), 10–18.
[16] M.A. Hall, Correlation-based feature subset selection for machine learning, *PhD. Thesis*, The University of Waikato, New Zealand, 1999.
[17] H. Hotelling, Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology* 24 (1933), 417-441.
[18] C.J. Jiang, C. Chen, J. Chang, R. Jan and T.C. Chiang, Construct small worlds in wireless networks using data mules, in: *Proc. of the IEEE Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'08)*, (2008), pp. 28-35.
[19] M. Juhola and M. Siermala, A scatter method for data and variable importance evaluation, *Integrated Computer-Aided Engineering* 19(2) (2012), 137-150.
[20] J.W. Kim, H.E. Jeon and J. Lee, Network Management Framework and Lifetime Evaluation Method for Wireless Sensor Networks, *Integrated Computer-Aided Engineering* 19(2) (2012), 165-178.
[21] J. Kleinberg, The small-world phenomenon: an algorithmic perspective, in: *Proc. of the ACM Symposium on Theory of Computing*, 2000, pp. 163 –170.
[22] I. Kononenko, Estimating attributes: analysis and extensions of RELIEF, in: *Proc. of the European Conference on Machine Learning*, 1994, pp. 171-182.
[23] A. Iamnitchi, M. Ripeanu and I. Foster, Small-world file-sharing communities. In: *Proc. 23rd Conference of the IEEE Communications Society (InfoCom'04)*, 2004, pp. 952-963.
[24] S. Lozano, A. Arenas and A. Sánchez, Mesoscopic structure conditions the emergence of cooperation on social networks. *PLoS ONE* 3(4) (2008), e1892.
[25] I. Marti, V.R. Tomas, L.A. García and J.J. Martínez, Multi-agent system for managing adverse weather situations on the road network. *Integrated Computer-Aided Engineering* 17(2) (2010), 145-155.
[26] A. Medina, A. Lakhina, I. Matta, J. Byers, BRITE: an approach to universal topology generation, in: *Proc. of the 9th. MASCOT Symposium*, 2001, pp. 346 -353.
[27] D.A. Mejia, J. Favela, A. Moran, Preserving interaction threads through the use of smartphones in hospitals, in: *Proc. of the 15th Int. Workshop on Groupware (CRIWG'09)*. LNCS 5784, Douro, Portugal, 2009, pp. 17-31.
[28] M.A. Nowak, Five rules for the evolution of cooperation, *Science* 314 (2006), 1560-1563.
[29] Oasis Consortium, Devices Profile for Web Services Version 1.1, *OASIS standard specification*, 2009.
[30] S.F. Ochoa, G. Bravo, J. Pino and J.F. Rodriguez, Coordinating loosely-coupled work in construction inspection activities. *Group Decision and Negotiation* 20(1) (2011), 39-56.
[31] K. Pawlikowski, H.-D.J. Jeong and J.-S.R Lee, On credibility of simulation studies of telecommunication networks, *IEEE Communications Magazine* 40(1) (2002), 132-139.
[32] D. Pinelle and C. Gutwin, Loose coupling and healthcare organizations: deployment strategies for groupware. *CSCW Journal* 15(5-6) (2006), 537-572.
[33] F. Ponci, L. Cristaldi, M. Faifer and M. Riva, Multi-agent systems: an example of power system dynamic reconfiguration, *Integrated Computer-Aided Engineering* 17(4) (2010), 359-372.
[34] W. Poundstone, Prisoner's dilemma, *Doubleday*, NY, USA, 1992.
[35] D. Qiu and R. Srikant, Modeling and performance analysis of BitTorrent-like peer-to-peer networks, in: *Proc. of SIGCOMM Conference*, 2004, pp. 367-378.
[36] N. Raveendranathan, S. Galzarano, V. Loseu, R. Gravina, R. Giannantonio, M. Sgroi, R. Jafari and G. Fortino, From Modeling to Implementation of Virtual Sensors in Body Sensor Networks, *IEEE Sensors Journal* 12(3) (2012), 583-593.
[37] S. Roy, H. Pucha, Z. Zhang, Y. Hu and L. Qiu, Overlay node placement: analysis, algorithms and impact on applications, in: *Proc. of the 27th. ICDCS Conference*, 53, 2007.
[38] F.C. Santos, J.F. Rodrigues and J.M. Pacheco, Graph topology plays a determinant role in the evolution of cooperation, in: *Proc. of Royal Society B: Biological Sc.* 273, 2006, pp. 51-55.
[39] R. Santos, J. Orozco and S.F. Ochoa, A real-time analysis approach in opportunistic networks. *ACM SIGBED Review* 8(3) (2011), 40-43.
[40] L. Sarmenta and S. Hirano, Bayanihan: building and studying web-based volunteer computing systems using Java, *Future Generation Computer Systems* 15 (1999), 675–686.
[41] D. Vega, Design and implementation of a simulator to explore cooperation in distributed environments. *Master thesis in Telecommunication Engineering & Management*, Universitat Politècnica de Catalunya, Spain, 2010.
[42] Y. Wang and A. Nakao, On cooperative and efficient overlay network evolution based on a group selection pattern. *IEEE Trans. on System Man Cyber. Part B* 40 (2010), 656-667.
[43] B.M. Waxman, Routing of multipoint connections, *IEEE J. on Selected Areas in Comm.* 6(9) (1988), 1617-1622.