Hedonic Coalition Formation for Distributed Task Allocation among Wireless Agents

Walid Saad[1], Zhu Han[2], Tamer Başar[3], Mérouane Debbah[4], and Are Hjørungnes[1]

[1]UNIK - University Graduate Center, University of Oslo, Norway, {saad,arehj}@unik.no.

[2]Electrical and Computer Engineering Department, University of Houston, USA, zhan2@mail.uh.edu.

[3]Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, USA, basar1@illinois.edu.

[4]Alcatel-Lucent Chair on Flexible Radio, SUPÉLEC, Gif-sur-Yvette, France, merouane.debbah@supelec.fr.

Abstract

Autonomous wireless agents such as unmanned aerial vehicles, mobile base stations, cognitive devices, or self-operating wireless nodes present a great potential for deployment in next generation wireless networks. While current literature has been mainly focused on the use of agents within robotics or software engineering applications, this paper proposes a novel usage model for self-organizing agents suited to wireless communication networks. In the proposed model, a number of agents are required to collect data from several arbitrarily located tasks. Each task represents a queue of packets that require collection and subsequent wireless transmission by the agents to a central receiver. The problem is modeled as a *hedonic coalition formation* game between the agents and the tasks that interact in order to form disjoint coalitions. Each formed coalition is modeled as a polling system consisting of a number of agents, designated as collectors, which move between the different tasks present in the coalition, collect and transmit the packets. Within each coalition, some agents might also take the role of a relay for improving the packet success rate of the transmission. The proposed hedonic coalition formation algorithm allows the tasks and the agents to take distributed decisions to join or leave a coalition, based on the achieved benefit in terms of effective throughput, and the cost in terms of polling system delay. As a result of these decisions, the agents and tasks structure themselves into independent disjoint coalitions which constitute a Nash-stable network partition. Moreover, the proposed coalition formation algorithm allows the agents and tasks to adapt the topology to environmental changes such as the arrival of new tasks, the removal of existing tasks, or the mobility of the tasks. Simulation results show how the proposed algorithm allows the agents and tasks to self-organize into independent coalitions, while improving the performance, in terms of average player (agent or task) payoff, of at least 30.26% (for a network of 5 agents with up to 25 tasks) relatively to a scheme that allocates nearby tasks equally among agents.

Keywords: Multiagent wireless networks, game theory, hedonic coalitions, task allocation, ad hoc networks.

I. INTRODUCTION

Next generation wireless networks will present a highly complex and dynamic environment characterized by a large number of heterogeneous information sources, and a variety of distributed network nodes. This is mainly due to the recent emergence of large-scale, distributed, and heterogeneous communication systems which are continuously increasing in size, traffic, applications, services, etc. For maintaining a satisfactory operation of such networks, there is a constant need for dynamically optimizing their performance, monitoring their operation and reconfiguring their topology. Due to the ubiquitous nature of such wireless networks, it is inherent to have self-organizing autonomous nodes (agents), that can service these networks at different levels such as data collection, monitoring, optimization, management, maintenance, among others [1–8]. These nodes belong to the authority maintaining the network, and must be able to survey large scale networks, and perform very specific tasks at different points in time, in a distributed and autonomous manner, with very little reliance on any centralized authority [1–3], [6–8].

While the use of such autonomous agents has been thoroughly investigated in robotics, computer systems or software engineering, research models that tackle the use of such agents in wireless communication networks are few. However, recently, the need for such agents in wireless networks has become of noticeable importance as many next generation networks encompass several wireless node types, such as cognitive devices or unmanned aerial vehicles, that are autonomous and self-adapting [1–8]. A key problem in this context is the problem of task allocation among a group of agents that need to execute a number of tasks. This problem has been already tackled in areas such as robotics control [9–11], or software systems [12], [13]. However, most of these existing models are unsuitable for task allocation problems in the context of wireless networks due to various reasons: (i)- The task allocation problems studied in the existing papers are mainly tailored for military operations, computer systems, or software engineering and, thus, cannot be readily applied in models pertaining to wireless networks, (ii)- the tasks are generally considered as static abstract entities with very simple characteristics and no intelligence (e.g. the tasks are just points in a plane) which is a major limitation, and (iii)- the existing models do not consider any aspects of wireless communication networks such as the characteristics of the wireless channel, the presence of data traffic, the need for wireless data transmission, or other wireless-specific specifications. In this context, numerous applications in next generation wireless networks require a number of agent-nodes to perform specific wireless-related tasks that emerge over time and are not pre-assigned. One example of these applications is the case where a number of wireless nodes are required to monitor the operation of the network or perform relaying at different times and locations [1], [2], [4–8]. In such applications, the objective is to provide algorithms that allow the agents to autonomously share the tasks among each other. The main existing contributions within wireless networking in this area [14–18],

are focused on deploying unmanned aerial vehicles (UAVs) which can act as self-deploying autonomous agents that can efficiently perform *pre-assigned* tasks in numerous applications such as connectivity improvement in ad hoc network [15], routing [16], [17], and medium access control [18]. However, these contributions focus on *centralized* solutions for specific problems such as finding the optimal locations for the deployment of UAVs or devising efficient routing algorithms in ad hoc networks in the presence of UAVs (one or more). Moreover, in these papers, the tasks that the agents must accomplish are *pre-assigned and pre-determined*. In contrast, as previously mentioned, many applications in wireless networks require the agents to autonomously assign the tasks among themselves. Hence, it is inherent to devise algorithms, in the context of wireless networks, that allow an autonomous and distributed task allocation process among a number of *wireless agents*[1] with little reliance on centralized entities.

The main contribution of this paper is to propose a novel wireless communication-oriented model for the problem of task allocation among a number of autonomous agents. The proposed model considers a number of wireless agents that are required to collect data from arbitrarily located tasks. Each task represents a source of data, i.e., a queue with a Poisson arrival of packets, that the agents must collect and transmit via a wireless link to a central receiver. This formulation is deemed suitable to model several problems in next generation networks such as video surveillance in wireless networks, self-deployment of mobile relays in IEEE 802.16j networks [2], data collection in ad hoc and sensor networks [8], operation of mobile base stations in vehicular ad hoc networks [6] and mobile ad hoc networks [7] (the so called *message ferry* operation), wireless monitoring of randomly located sites, autonomous deployment of unmanned air vehicles in military ad hoc networks, and many other applications. For allocating the tasks, we introduce a novel framework from coalitional game theory, known as *hedonic coalition formation*. Albeit hedonic games have been widely used in game theory, to the best of our knowledge, no existing work utilized this framework in a communication or wireless environment. Thus, we model the task allocation problem as a hedonic coalition formation game between the agents and the tasks, and we introduce an algorithm for forming coalitions. Each formed coalition is modeled as a polling system consisting of a number of agents, designated as *collectors*, which act as a single server that moves continuously between the different tasks (queues) present in the coalition, gathering and transmitting the collected packets to a common receiver. Further, within each coalition, some agents can act as *relays* for improving the packet success rate during the wireless transmission. For forming coalitions, the agents and tasks can autonomously make a decision to join or leave a coalition based on well defined

---

[1]The term *wireless agent* refers to any node that can act autonomously and can perform wireless transmission. Examples of wireless agents are unmanned aerial vehicles [15], mobile base stations [6–8], cognitive wireless devices [3], or self-deploying mobile relay stations [2].

individual preference relations. These preferences are based on a coalitional value function that takes into account the benefits received from servicing a task, in terms of effective throughput (data collected), as well as the cost in terms of the polling system delay incurred from the time needed for servicing all the tasks in a coalition. We study the properties of the proposed algorithm, and show that it always converges to a Nash-stable network partition. Further, we investigate how the network topology can self-adapt to environmental changes such as the deployment of new tasks, the removal of existing tasks, and low mobility of the tasks. Simulation results show how the proposed algorithm allows the network to self-organize, while ensuring a significant performance improvement, in terms of average player (task or agent) payoff, compared to a scheme that assigns nearby tasks equally among the agents.

The remainder of this paper is organized as follows: Section II presents and motivates the proposed system model. In Section III, we model the task allocation problem problem as a transferable utility coalitional game and propose a suited utility function. In Section IV, we classify the task allocation coalitional game as a hedonic coalition formation game, we discuss its key properties and we introduce the algorithm for coalition formation. Simulation results are presented, discussed, and analyzed in Section V. Finally, conclusions are drawn in Section VI.

II. SYSTEM MODEL

Consider a network consisting of M wireless agents that belong to a single network operator and that are controlled by a central command center (e.g., a central controller node or a satellite system). These agents are required to service T tasks that are arbitrarily located in a geographic area having an associated central wireless receiver connected to the command center. The tasks are entities that, in general, belong to one or more independent owners[2]. These tasks' owners can be, for example, service providers or third party operators. We denote the set of agents, and tasks by $\mathcal{M} = \{1, \ldots, M\}$, and $\mathcal{T} = \{1, \ldots, T\}$, respectively. In this paper, we consider only the case where the number of tasks is larger than the number of agents, hence, $T > M$. The main motivation behind this consideration is that, for most networks, the number of agents assigned to a specific area is generally small due to cost factors for example. Each task $i \in \mathcal{T}$ represents an M/D/1 queueing system[3], whereby packets of constant size B are generated using a Poisson arrival with an average arrival rate of λ_i. Hence, in the proposed model, we consider different classes of tasks each having its corresponding λ_i. The tasks we consider are sources of data that cannot send their information to the central receiver (and, subsequently, to the command center) without the help of an agent. These tasks can represent a group of

[2]The scenario where all tasks and agents are owned by the same entity is a particular case of this generic model.

[3]Other types of queues, e.g., M/M/1, can also be considered without loss of generality in the coalition formation process proposed in this paper.

mobile devices, such as sensors, video surveillance devices, or any other static or dynamic wireless nodes that have limited power and are unable to provide long-distance transmission. Hence, these devices (tasks) need to buffer their data locally and await to be serviced by an agent that can collect the data. For example, an agent such as a mobile base station, a mobile relay or a UAV can provide a line-of-sight link that can facilitate the transmission from the tasks to the central receiver. The tasks can also be mapped to any other source of packet data that require collection by an agent for transmission[4]. For servicing a task, each agent is required to move to the task location, collect the data, and transmit it using a wireless link to the central receiver. The command center periodically downloads this data from the receiver, e.g., through a backbone network. Each agent $i \in \mathcal{M}$ offers a link transmission capacity of μ_i, in packets/second, with which the agent can service the data from any task. Hence, the quantity $\frac{1}{\mu_i}$ would represent the well known service time for a single packet that is being serviced by agent i. The agent which is collecting the data from a task is referred to as *collector*. In addition, each agent $i \in \mathcal{M}$ can transmit the data to the receiver with a maximum transmit power of $P_i = \tilde{P}$, assumed the same for all agents [5].

The proposed model allows each task to be serviced by multiple agents, and also, each agent (or group of agents) to service multiple tasks. Whenever a task is serviced by multiple agents, each agent can act as either a *collector* or a *relay*. Any group of agents that act together for data collection from the same task, can be seen as a single collector with improved link transmission capacity. In this paper, we consider that the link transmission capacity depends solely on the capabilities of the agents and not on the nature of the tasks. In this context, given a group of agents $G \subseteq \mathcal{M}$ that are acting as collectors for any task, the total link transmission capacity with which tasks can be serviced with by G can be given by

$$\mu^G = \sum_{j \in G} \mu_j. \tag{1}$$

For forming a single collector, multiple agents can easily coordinate the data extraction, and then transmission from every task, so as to allow a larger link transmission capacity for the serviced task as per (1). Moreover, the transmission of the packets by the agents from a task $i \in \mathcal{T}$ to the central receiver is subject to packet loss due to the fading on the wireless channel. In this regard, in addition to acting as collectors, some agents may act as *relays* for a task. These relay-agents would locate themselves at equal distances from the task (given that the task is already being served by *at least one* collector), and, hence, the collectors transmit the data to the receiver through multi-hop agents, improving the probability of successful transmission. In

---

[4]The tasks can be moving with a periodic low mobility, as will be seen in later sections.

[5]Note that, different maximum transmit power values can be easily accommodated in the coalition formation algorithm proposed in the next sections.
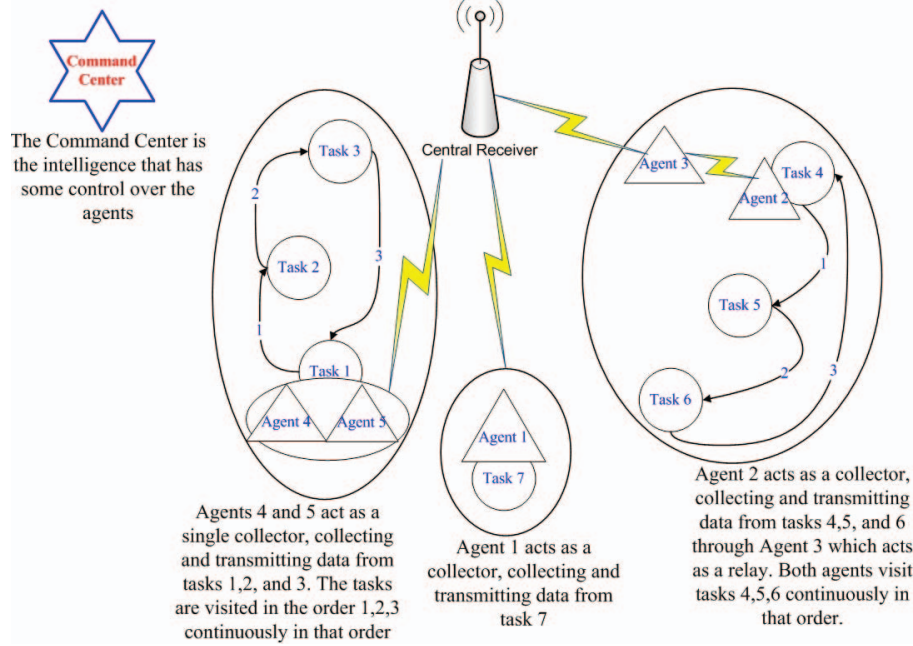
Fig. 1. An illustrative example of the proposed model for task allocation in wireless networks (the agents are dynamic, i.e., they move from one task to the other continuously).

this context, in Rayleigh fading, the probability of successful transmission of a packet of size B bits from the collectors present at a task $i \in \mathcal{T}$ through a path of m agents, $Q_i = \{i_1, \ldots, i_m\}$, where $i_1 = i$ is the task being serviced, i_m is the central receiver (CR), and any other $i_h \in Q_i$ is a *relay*-agent, can be given by

$$\mathrm{Pr}_{i,\mathrm{CR}} = \prod_{h=1}^{m-1} \mathrm{Pr}_{i_h,i_{h+1}}^{B}, \tag{2}$$

where $\mathrm{Pr}_{i_h,i_{h+1}}$ is the the probability of successful transmission of a single bit from agent i_h to agent (or the central receiver) i_{h+1}. This probability can be given by the probability of maintaining the SNR at the receiver above a target level ν_0 as follows [19]

$$\mathrm{Pr}_{i,i_{h+1}} = \exp\left(-\frac{\sigma^2 \nu_0 (D_{i_h,i_{h+1}})^\alpha}{\kappa \tilde{P}}\right), \tag{3}$$

where σ^2 is the variance of the Gaussian noise, κ is a path loss constant, α is the path loss exponent, $D_{i_h,i_{h+1}}$ is the distance between nodes i_h and i_{h+1}, and $\tilde{P}$ is the maximum transmit power of agent i_h.

For servicing a number of tasks $C \subseteq \mathcal{T}$, a group of agents $G \subseteq \mathcal{M}$ (collectors and relays) can sequentially move from one task to the other in C with a constant velocity η. The group G of agents, servicing the tasks in C, stop at each task, with the collectors collecting and transmitting the packets using the relays (if any). The collectors would move from one task to the other, only if all the packets in the queue at the current task have been transmitted to the receiver (the process through which the agents move from one task to the other for data collection is cyclic). Simultaneously with the collectors, the relays also move, positioning themselves

at equal distances on the line connecting the task being currently served by the collectors, and the central receiver. With this proposed model, the final network will consist of groups of tasks serviced by groups of agents, continuously. An illustration of this model is shown in Figure 1.

Consequently, given this proposed model, the main objective is to provide an algorithm for distributing the tasks between the agents, given the operation of the agents previously described and shown in Figure 1. In other words, the main goal is to allow the agents and task to autonomously form the coalitional structure in Figure 1 and adapt it to environmental changes. For this purpose, the following sections formulate a game theoretic approach for achieving this objective.

III. COALITIONAL GAME FORMULATION

In this section, we model the proposed problem as a coalitional game with transferable utility, and we propose a suitable utility function.

A. Game Formulation

By inspecting Figure 1, one can clearly see that the task allocation problem among the agents can be mapped into the problem of the formation of coalitions. In this regard, coalitional game theory [20, Ch. 9] provides a suitable analytical tool for studying the formation of cooperative groups, i.e., coalitions, among a number of players. For the proposed model, the coalitional game is played between the agents and the tasks. Hence, the players set for the proposed task allocation coalitional game is denoted by $\mathcal{N}$, and contains both agents and tasks, i.e., $\mathcal{N} = \mathcal{M} \cup \mathcal{T}$. In the remainder of this paper, we use the term *player* to indicate either a task or an agent.

For any coalition $S \subseteq \mathcal{N}$ containing a number of agents and tasks, the agents belonging to this coalition can structure themselves into collectors and relays. Subsequently, as explained in the previous section, within each coalition, the collector-agents will continuously move from one task to the other, stopping at each task, and transmitting all the packets available in the queue to the central receiver, through the relay-agents (if any). This proposed task servicing scheme can be mapped to a well-known concept that is ubiquitous in computer systems, which is the concept of a *polling system* [21]. In a polling system, a single server moves between multiple queues in order to extract the packets from each queue, in a sequential and cyclic manner. Models pertaining to polling systems have been widely developed in various disciplines ranging from computer systems to communication networks, and different strategies for servicing the queues exist [21–24]. In the proposed task allocation model, the collectors of every coalition in our model are considered as a single server that is servicing the tasks (queues) sequentially, in a cyclic manner, i.e., after servicing the last task in a coalition $S \subseteq \mathcal{N}$, the collectors of S return to the first task in S that they previously visited hence repeating their route continuously. Further, whenever the collectors stop at any task $i \in S$, they collect and transmit the

data present at this task until the queue is empty. This method of allowing the server to service a queue until emptying the queue is known as the *exhaustive* strategy for a polling system, which is applied at the level of every coalition $S \subseteq \mathcal{N}$ in our model. Moreover, the time for the server to move from one queue to the other is known as the *switchover* time. Consequently, we highlight the following property:

Property 1: In the proposed task allocation model, every coalition $S \subseteq \mathcal{N}$ is a polling system with an *exhaustive* polling strategy and deterministic non-zero *switchover* times. In each such polling system S, the collector-agents are seen as the polling system server, and the tasks are the queues that the collector-agents must service.

Furthermore, for any coalition S, once the queue at a task $i \in S$ is emptied, the collectors and relays in a coalition move from task i to the next task $j \in S$ with a constant velocity η, hence, incurring a switchover time $\theta_{i,j}$. The switchover time in our model corresponds to the time it takes for all the agents (collectors and relays) to move from one task to the next, which, assuming all agents start their mobility at the same time, maps to the time needed for the farthest agent to move from one task to the next. Since we consider only straight line trajectories for collectors and relays, and due to the fact that the relays always position themselves at equal distances on the line connecting the tasks in a coalition to the receiver, we have the following property (clearly seen through the geometry of Figure 1).

Property 2: Within any given coalition S, the switchover time between two tasks corresponds to the *constant* time it takes for one of the *collectors* to move from one of the tasks to the next.

Having modeled every coalition $S \subseteq \mathcal{N}$ as a polling system, we investigate the average delay incurred per coalition. In fact, for polling systems, finding exact expressions for the delay at every queue is a highly complicated task, and hence, no general closed-form expressions for the delay at every queue in a polling system can be found [21], [22][6]. In this regard, a key criterion used for the analysis of the delay incurred by a polling system is the *pseudo-conservation* law which provides closed-form expressions for weighted sum of the means of the waiting times at the queues [21], [22]. For providing the pseudo-conservation law for a coalition $S \subseteq \mathcal{N}$ composed of a number of agents and a number of tasks, we make the following definitions. First, within coalition S, a group of agents $G_S \subseteq S \cap \mathcal{M}$ are designated as collectors. Second, for each task $i \in S \cap \mathcal{T}$ with an average arrival rate of λ_i, and served by a number of collectors $|G_S|$ with a link transmission capacity of μ^{G_S} (as given by (1)), we define the utilization factor of task i $\rho_i = \frac{\lambda_i}{\mu^{G_S}}$. Further, we define $\rho_S \triangleq \sum_{i \in S \cap \mathcal{T}} \rho_i$. Given these definitions, for a coalition S, the weighted sum of the means of

---

[6]Note that some approximations [22] exist for polling systems under heavy traffic or large switchover times, but in our problem, they are not suitable as we require a more general delay expression.

the waiting times by the agents at all the tasks in the coalition are given by the pseudo-conservation law as follows [22, Section. VI-B] (taking into account that our switchover and service times are deterministic)

$$\sum_{i \in S \cap \mathcal{T}} \rho_i \bar{W}_i = \rho_S \frac{\sum_{i \in S \cap \mathcal{T}} \frac{\rho_i}{\mu^{GS}}}{2(1 - \rho_S)} + \rho_S \frac{\theta_S^2}{2} + \frac{\theta_S}{2(1 - \rho_S)} \left[\rho_S^2 - \sum_{i \in S \cap \mathcal{T}} \rho_i^2 \right], \tag{4}$$

where $\bar{W}_i$ is the mean waiting time at task i and $\theta_S = \sum_{h=1}^{|S \cap \mathcal{T}|} \theta_{i_h, i_{h+1}}$ is the sum of the switchover times given a path of tasks $\{i_1, \ldots, i_{|S \cap \mathcal{T}|}\}$ followed by the agents, with $i_h \in S \cap \mathcal{T}$, $\forall h \in \{1, \ldots, |S \cap \mathcal{T}|\}$ and $i_{|S \cap \mathcal{T}|+1} = i_1$. The first term in the right hand side of (4) is the well known expression for the average queueing delay for M/D/1 queues, weighed by ρ_S. The second and third terms in the right hand side of (4) represent the average delay increase incurred by the travel time required for the collectors to move from one task to the other, i.e., the delay resulting from the switchover period. Further, for any coalition S that must form in the system, the following condition must hold:

$$\rho_S < 1. \tag{5}$$

This condition is a requirement for the stability of any polling system [21–24] and, thus, must be satisfied for any coalition that will form in the proposed model. In the event where this condition is violated, the system is considered unstable and the delay is considered as infinite. In this regard, the analysis presented in the remainder of this paper will take into account this condition and its impact on the coalition formation process (as demonstrated in the next sections, a coalition where $\rho_S \geq 1$ will never form).

B. Utility Function

In the proposed game, for every coalition $S \subseteq \mathcal{N}$, the agents must determine the order in which the tasks in S are visited, i.e., the path $\{i_1, \ldots, i_{|S \cap \mathcal{T}|}\}$ which is an ordering over the set of tasks in S given by $S \cap \mathcal{T}$. Naturally, the agents must select the path that minimizes the total switchover time for one round of data collection. This can be mapped to the following well-known problem:

Property 3: The problem of finding the path that minimizes the total switchover time for one round of data collection within a coalition $S \subseteq \mathcal{N}$ is mapped into the *traveling salesman* problem [25], where a salesman, i.e., the agents $S \cap \mathcal{M}$, is required to minimize the time of visiting a series of cities, i.e., the tasks $S \cap \mathcal{T}$.

It is widely known that the solution for the traveling salesman problem is NP-complete [25], and, hence, there has been numerous heuristic algorithms for finding an acceptable near-optimal solution. One of the simplest of such algorithms is the nearest neighbor algorithm (also known as the greedy algorithm) [25]. In this algorithm, starting from a given city the salesman chooses the closet city as his next visit. Using the nearest neighbor algorithm, the ordering of the cities which minimizes the overall route is selected. The

nearest neighbor algorithm is sub-optimal, however, it can quickly find a near-optimal solution (in most cases) and its computational complexity is small (linear in the number of cities) [25], hence making it suitable for complicated problems such as the task allocation problem we are considering. Therefore, in the proposed model, for every coalition S, the agents can easily work out the nearest neighbor route for the tasks, and operate according to it.

Having modeled each coalition as a polling system, the pseudo-conservation law in (4) allows to evaluate the cost, in terms of average waiting time (or delay), from forming a particular coalition. However, for every coalition, there is a benefit, in terms of the average effective throughput that the coalition is able to achieve. The average effective throughput for a coalition S is given by

$$L_S = \sum_{i \in S \cap \mathcal{T}} \lambda_i \cdot \mathrm{Pr}_{i,\mathrm{CR}}, \tag{6}$$

with $\mathrm{Pr}_{i,\mathrm{CR}}$ given by (2). By closely inspecting (1), one can see that adding more collectors improves the transmission link capacity, and, thus, reduces the service time that a certain task perceives. Based on this property and by using (4) one can easily see that, adding more collectors, i.e., improving the service time, reduces the overall delay in (4) [21–24]. Further, adding more relays would reduce the distance over which transmission is occurring, thus, improving the probability of successful transmission as per (2) [6], [19]. In consequence, using (6), one can see that this improvement in the probability of successful transmission is translated into an improvement in the effective throughput. Hence, each agent role (collector or relay) possesses its own benefit for a coalition.

A suitable criterion for characterizing the utility in networks that exhibit a tradeoff between the throughput and the delay is the concept of system *power* which is defined as the ratio of some power of the throughput and the delay (or a power of the delay) [26]. Hence, the concept of power is an attractive notion that allows to capture the fundamental tradeoff between throughput and delay in the proposed task allocation model. Power has been used thoroughly in the literature in applications that are sensitive to throughput as well as delay [27–29]. Mainly, for the proposed game, the utility of every coalition S is evaluated using a coalitional value function based on the power concept from [29] as follows

$$v(S) = \begin{cases} \delta \dfrac{L_S^{\beta}}{(\sum_{i \in S \cap \mathcal{T}} \rho_i \bar{W}_i)^{(1-\beta)}}, & \text{if } \rho_S < 1 \text{ and } |S| > 1, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\beta \in (0,1)$ is a throughput-delay tradeoff parameter. In (7), the term δ represents the price per unit power that the network offers to coalition S. Hence, δ represents a generic control parameter that allows the network operator to somehow monitor the behavior of the players. The use of such control parameters

is prevalent in game theory [30–34]. In certain scenarios, δ would represent physical monetary values paid by the operator to the different entities (agents and tasks). In such a case, on one hand, for the tasks, the operator simply would pay the tasks' owners for the amount of data (and its corresponding quality as per (7)) each one of their tasks had generated. On the other hand, for the agents, the payment would, for example, represent either a reward for the behavior of the agents or the proportion of maintenance or servicing that each agent would receiver from its operator. In this sense, the utility function in (7) would, thus, represents the total revenue achieved by a coalition S, given the network power that coalition S obtains. For coalitions that consist of a single agent or a single task, i.e., coalitions of size 1, the utility assigned is 0 due to the fact that such coalitions generate no benefit for their member (a single agent can collect no data unless it moves to task, while a single task cannot transmit any of its generated data without an agent collecting this data). Further, any coalition where condition (5) is not satisfied is also given a zero utility, since, in this case, the polling system that the coalition represents is unstable, and hence has an infinite delay.

Consequently, given the set of players $\mathcal{N}$, and the value function given in (7), we define a coalitional game $(\mathcal{N}, v)$ with transferable utility (TU). The utility in (7) represents the amount of money or revenue received by a coalition, and, hence, this amount can be arbitrarily apportioned between the coalition members, which justifies the TU nature of the game. For dividing this utility between the players, we adopt the equal fair allocation rule, whereby the payoff of any player $i \in S$, denoted by x_i^S is given by

$$x_i^S = \frac{v(S)}{|S|}. \tag{8}$$

The payoff x_i^S represents the amount of revenue that player $i \in S$ receives from the total revenue $v(S)$ that coalition S generates. The main motivation behind adopting the equal fair allocation rule is in order to highlight the fact that the agents and the tasks value each others equally. As seen in (7), the presence of an agent in a coalition is crucial in order for the tasks to obtain any payoff, and, vice versa, the presence of a task in a coalition is required for the agent to be able to obtain any kind of utility. Nonetheless, the proposed model and algorithm can accommodate any other type of payoff allocation rule. For instance, although in traditional coalitional games, the allocation rule may have a strong impact on the game's solution, for the proposed game, other allocation rules can be used with little impact on the analysis that is presented in the rest of the paper. This is mainly due to the nature and class of the proposed game which is quite different from traditional coalitional games. In fact, as clearly seen from (4) and (7), whenever the number of tasks in a coalition increases, the total delay increases, hence, reducing the utility from forming a coalition. Further, in a coalition where the number of tasks is large, the condition of stability for the polling system, as given by (5), can be easily violated due to heavy traffic incoming from a large number of tasks, thus, yielding a zero utility

as per (7). Hence, forming coalitions between the tasks and the agents entails a cost that can limit the size of a coalition. In this regard, traditional solution concepts for TU games, such as the core [20], may not be applicable. In fact, in order for the core to exist, as a solution concept, a TU coalitional game must ensure that the grand coalition, i.e., the coalition of all players will form. However, as seen in Figure 1 and corroborated by the utility in (7), in general, due to the cost for coalition formation, the grand coalition will not form. Instead, independent and disjoint coalitions appear in the network as a result of the task allocation process. In this regard, the proposed game is classified as a *coalition formation game* [30–34], and the objective is to find an algorithm that allows to form the coalition structure, instead of finding only a solution concept, such as the core, which aims mainly at stabilizing a grand coalition of all players.

IV. TASK ALLOCATION AS A HEDONIC COALITION FORMATION GAME

In this section, we map the proposed task allocation problem to a hedonic coalition formation game with an underlying transferable utility, and we propose a distributed algorithm for forming the coalitions using concepts from hedonic games.

A. Hedonic Coalition Formation: Concepts and Model

As already mentioned, the proposed task allocation model entails the formation of disjoint coalitions, and, hence, the proposed game is classified as a coalition formation game. In fact, coalition formation has been a topic of high interest in game theory [30–34]. Notably, in [32–34], a class of coalition formation games known as *hedonic coalition formation games* is investigated. This class of games entails several interesting properties that can be applied, not only in economics such as in [32–34], but also in wireless networks as we will demonstrate in this paper. The two key requirements for classifying a coalitional game as a *hedonic* game are as follows [32]:

Condition 1: - (Hedonic Conditions) - A coalition formation game is classified as hedonic if

1) The payoff of any player depends *solely* on the members of the coalition to which the player belongs.

2) The coalitions form as a result of the *preferences* of the players over their possible coalitions' set.

These two conditions characterize the framework of hedonic games. Mainly, the term *hedonic* pertains to the first condition above, whereby the payoff of any player i, in a hedonic game, must depend only on the identity of the players in the coalition to which player i belongs, with no dependence on the other players. For the second condition, in the remainder of this section, we will formally define how the preferences of the players over the coalitions can be used for the formation process.

Prior to investigating the application of hedonic games in the proposed model, we introduce some definitions, taken from [32].

Definition 1: A *coalition structure* or a *coalition partition* is defined as the set $\Pi = \{S_1, \ldots, S_l\}$ which partitions the players set $\mathcal{N}$, i.e., $\forall \, k$, $S_k \subseteq \mathcal{N}$ are disjoint coalitions such that $\cup_{k=1}^{l} S_k = \mathcal{N}$ (an example of a partition Π is shown in Figure 1).

Definition 2: Given a partition Π of $\mathcal{N}$, for every player $i \in \mathcal{N}$ we denote by $S_\Pi(i)$, the coalition to which player i belongs, i.e., coalition $S_\Pi(i) = S_k \in \Pi$, such that $i \in S_k$.

In a hedonic game setting, each player must build preferences over its own set of possible coalitions. In other words, each player must be able to compare the coalitions, and order them based on which coalition the player prefers being a member of. For evaluating these preferences of the players over the coalitions, we define the concept of a preference relation or order as follows [32]:

Definition 3: For any player $i \in \mathcal{N}$, a *preference relation* or *order* $\succeq_i$ is defined as a complete, reflexive, and transitive binary relation over the set of all coalitions that player i can possibly form, i.e., the set $\{S_k \subseteq \mathcal{N} : i \in S_k\}$.

Consequently, for a player $i \in \mathcal{N}$, given two coalitions $S_1 \subseteq \mathcal{N}$ and, $S_2 \subseteq \mathcal{N}$ such that $i \in S_1$ and $i \in S_2$, $S_1 \succeq_i S_2$ indicates that player i prefers to be part of coalition S_1, over being part of coalition S_2, or at least, i prefers both coalitions equally. Further, using the asymmetric counterpart of $\succeq_i$, denoted by $\succ_i$, then $S_1 \succ_i S_2$, indicates that player i *strictly* prefers being a member of S_1 over being a member of S_2. For every application, an adequate preference relation $\succeq_i$ can be defined to allow the players to quantify their preferences. The preference relation can be a function of many parameters, such as the payoffs that the players receive from each coalition, the weight each player gives to other players, and so on. Given the set of players $\mathcal{N}$, and a preference relation $\succeq_i$ for every player $i \in \mathcal{N}$, a hedonic coalition formation game is formally defined as follows [32]:

Definition 4: A hedonic coalition formation game is a coalitional game that satisfies the two hedonic conditions previously prescribed, and is defined by the pair $(\mathcal{N}, \succ)$ where $\mathcal{N}$ is the set of players ($|\mathcal{N}| = N$), and $\succ$ is a *profile of preferences*, i.e., preference relations, $(\succeq_1, \ldots, \succeq_N)$ defined for every player in $\mathcal{N}$.

Having laid out and defined the main components of hedonic coalition formation games, we utilize this framework in order to provide a suitable solution for the task allocation problem proposed in Section II. For instance, the proposed task allocation problem is easily modeled as a $(\mathcal{N}, \succ)$ hedonic game, where $\mathcal{N}$ is the set of agents and tasks previously defined, and $\succ$ is a profile of preferences that we will shortly define. First and foremost, for the proposed game model, given a network partition Π of $\mathcal{N}$, the payoff of any player i, depends only on the identity of the members of the coalition to which i belongs. In other words, the payoff

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON MOBILE COMPUTING

14

of any player i depends solely on the players in the coalition in which player i belongs $S_\Pi(i)$ (easily seen through the formulation of Section III). Hence, our game verifies the first hedonic condition.

Furthermore, for modeling the task allocation problem as a hedonic coalition formation game, the preference relations of the players must be clearly defined. In this regard, we define two types of preference relations, a first type suited for indicating the preferences of the agents, and a second type suited for the tasks. Subsequently, for evaluating the preferences of any agent $i \in \mathcal{M}$, we define the following operation (this preference relation is common for all agents, hence we denote it by $\succeq_i = \succeq_\mathcal{M}, \ \forall i \in \mathcal{M}$)

$$S_2 \succeq_\mathcal{M} S_1 \Leftrightarrow u_i(S_2) \geq u_i(S_1), \tag{9}$$

where $S_1 \subseteq \mathcal{N}$ and $S_2 \subseteq \mathcal{N}$ are any two coalitions that contain agent i, i.e., $i \in S_1$ and $i \in S_2$ and $u_i : 2^\mathcal{N} \to \mathbb{R}$ is a preference function defined for any agent i as follows

$$u_i(S) = \begin{cases} \infty, & \text{if } S = S_\Pi(i) \ \& \ S \setminus \{i\} \subseteq \mathcal{T}, \\ 0, & \text{if } S \in h(i), \\ x_i^S. & \text{otherwise,} \end{cases} \tag{10}$$

where Π is the *current* coalition partition which is in place in the game, x_i^S is the payoff received by player i from any division of the value function among the players in coalition S such as the equal fair division given in (8), and $h(i)$ is the history set of player i. At any point in time, the history set $h(i)$ is a set that contains the coalitions that player i was a part of in the past, prior to the formation of the current partition Π. Note that, by using the defined preference relation, the players can compare any two coalitions S_1 and S_2 independently of whether these two coalitions belong to Π or not.

The main rationale behind the preference function u_i in (10) is as follows. In this model, the agents, being entities owned by the operator, seek out to achieve two conflicting objectives: (i)- Service all tasks in the network for the benefit of the operator, and (ii)- Maximize the quality of service, in terms of power as per (7), for extracting the data from the tasks. The preference function u_i must be able to allow the agents to make coalition formation decisions that can capture this tradeoff between servicing all tasks (at the benefit of the operator) and achieving a good quality of service (at the benefit of both agents and operator). For this purpose, as per (10), any agent i that is the *sole* agent servicing tasks in its current coalition $S = S_\Pi(i)$ such that $S_\Pi(i) \cap \mathcal{M} = \{i\}$, assigns an infinite preference value to S. Hence, in order to service all tasks, the agent always assigns a maximum preference to its current coalition, if this current coalition is composed of only tasks and does not contain other agents. This case of the preference function u_i forbids the agent from leaving a group of tasks, already assigned to it, unattended by other agents. In this context, this condition

pertains to the fist objective (objective (i) previously mentioned) of the agents and it implies that, whenever there is a risk of leaving tasks without service, the agent do not act selfishly, in contrast, they act in the benefit of the operator and remain with these tasks, independent of the payoff generated by these tasks. Such a decision allows an agent to avoid making a decision that can incur a risk of ultimately having tasks with no service in the network, in which case, the network operator would lose revenue from these unattended tasks and it may, for example, decide to replace the agent that led to the presence of such a group of tasks with no service. Otherwise, the agents' preference relation u_i would highlight the second objective of the agent, i.e., maximize its own payoff which maps into the revenue generated from the quality of service, i.e., the power as per (7). with which the tasks are being serviced. In this case, the preference relation is easily generated by the agents by comparing the value of the payoffs they receive from the two coalitions S_1 and S_2. Further, we note that no agent has any incentive to revisit a coalition that it had left previously, and hence, the agents assign a preference value of 0 for any coalition in their history (this can be seen as a basic learning process). In summary, taking into account the conflicting goals of the agents, between two coalitions S_1 and S_2, an agent i prefers the coalition that gives the better payoff, given that the agent is not alone in its current coalition, and the coalition with a better payoff is not in the history of agent i.

For the preferences of the tasks, an analogous approach can be taken. Formally, for evaluating the preferences of any task $j \in \mathcal{T}$, we define the following operation (this preference relation is common for all tasks, hence, we denote it by $\succeq_j = \succeq_{\mathcal{T}}, \ \forall j \in \mathcal{T}$)

$$S_2 \succeq_{\mathcal{T}} S_1 \Leftrightarrow w_j(S_2) \geq w_j(S_1), \tag{11}$$

with the tasks' preference function w_j defined as follows.

$$w_j(S) = \begin{cases} 0, & \text{if } S \in h(j), \\ x_j^S, & \text{otherwise.} \end{cases} \tag{12}$$

The preferences of the tasks are easily captured using the function w_j. The preference function of the tasks is different from that of the agents since the tasks are, in general, independent entities that act solely in their own interest. Thus, based on (12), each task prefers the coalition that provides the larger payoff x_j^S unless this coalition was already visited previously and left. In that case, the preference function of the tasks assigns a preference value of 0 for any coalition that the task had already visited in the past (and left to join another coalition). Using this preference relation, every task can evaluate its preferences over the possible coalitions that the task can form.

Consequently, the proposed task allocation model verifies both hedonic conditions, and, hence, we have the following:

Property 4: The proposed task allocation problem among the agents is modeled as a $(\mathcal{N}, \succ)$ hedonic coalition formation game, with the preference relations given by (9) and (11).

Note that the preference relations in (9) and (11) are also dependent on the underlying TU coalitional game described in Section III. Having formulated the problem as a hedonic game, the final task is to provide a distributed algorithm, based on the defined preferences, for forming the coalitions.

Prior to deriving the algorithm for coalition formation, we highlight the following proposition of the proposed hedonic coalition formation model as a result of the polling system stability indicated in (5)

Proposition 1: For the proposed hedonic coalition formation model for task allocation, assuming that all collector-agents have an equal link transmission capacity $\mu_i = \mu$, $\forall i \in \mathcal{M}$, any coalition $S \subseteq \mathcal{N}$ with $|S \cap \mathcal{M}|$ agents, must have at least $|G_S|_{\min}$ collector-agents $(G_S \subseteq S \cap \mathcal{M})$ as follows

$$|G_S| > |G_S|_{\min} = \frac{\sum_{i \in S \cap \mathcal{T}} \lambda_i}{\mu}. \tag{13}$$

Further, when all the tasks in S belong to the same class, we have

$$|G_S|_{\min} = \frac{|S \cap \mathcal{T}| \cdot \lambda}{\mu}, \tag{14}$$

which constitutes an upper bound on the number of collector-agents as a function of the number of tasks $|S \cap \mathcal{T}|$ for a given coalition S.

Proof: As per the defined preference relations in (10) and (12), any coalition that will form in the proposed model must be stable since no agent or task has an incentive to join an unstable coalition, hence, we have, for every coalition $S \subseteq \mathcal{N}$ having $|G_S|$ collectors with $G_S \subseteq S \cap \mathcal{M}$, we have from (5) $\rho_S < 1$, and thus

$$\sum_{i \in S \cap \mathcal{T}} \frac{\lambda_i}{\mu^{G_S}} < 1,$$

which, given the assumption that $\mu_i = \mu, \forall i \in \mathcal{M}$ yields

$$\frac{1}{|G_S| \cdot \mu} \cdot \sum_{i \in S \cap \mathcal{T}} \lambda_i < 1,$$

which yields

$$|G_S| > |G_S|_{\min} = \frac{\sum_{i \in S \cap \mathcal{T}} \lambda_i}{\mu}. \tag{15}$$

Further, if we assume that all the tasks belong to the same class, hence, $\lambda_i = \lambda$, $\forall i \in S \cap \mathcal{T}$, we immediately get, from (15)

$$|G_S|_{\min} = \frac{|S \cap \mathcal{T}| \cdot \lambda}{\mu}. \tag{16}$$

∎

Consequently, for any proposed coalition formation algorithm, the bounds on the number of collector-agents in any coalition S as given by Proposition 1 will always be satisfied.

B. Hedonic Coalition Formation: Algorithm

In the previous subsection, we modeled the task allocation problem as a hedonic coalition formation game. Having laid out the main building blocks, the remaining objective is to devise an algorithm for forming the coalitions. While literature that studies the characteristics of existing partitions in hedonic games, such as in [32–34], is abundant, the problem of forming the coalitions both in the hedonic and non-hedonic setting is a challenging problem [30]. In this paper, we introduce an algorithm for coalition formation that allows the players to make *selfish* decisions as to which coalitions they decide to join at any point in time. The proposed algorithm will exploit the concepts of the hedonic game model formulated in the previous subsection.

In this regard, for forming coalitions between the tasks and the agents, we propose the following rule for coalition formation:

Definition 5: Switch Rule - Given a partition $\Pi = \{S_1, \ldots, S_l\}$ of the set of players (agents and tasks) $\mathcal{N}$, a Player i decides to leave its current coalition $S_\Pi(i) = S_m$, for some $m \in \{1, \ldots, l\}$ and join another coalition $S_k \in \Pi \cup \{\emptyset\}$, $S_k \neq S_\Pi(i)$, if and only if $S_k \cup \{i\} \succ_i S_\Pi(i)$. Hence, $\{S_m, S_k\} \rightarrow \{S_m \setminus \{i\}, S_k \cup \{i\}\}$.

Through a single switch rule made by any player i, any current partition Π of $\mathcal{N}$ is transformed into $\Pi' = (\Pi \setminus \{S_m, S_k\}) \cup \{S_m \setminus \{i\}, S_k \cup \{i\}\}$. In simple terms, for every partition Π, the switch rule provides a mechanism through which any task or agent, can leave its current coalition $S_\Pi(i)$, and join another coalition $S_k \in \Pi$, given that the new coalition $S_k \cup \{i\}$ is strictly preferred over $S_\Pi(i)$ through any preference relation that i is using (in particular using the preference relations defined in (9) and (11)). Independent of the preference relations selected, the switch rule can be seen as a *selfish* decision made by a player, to move from its current coalition to a new coalition, regardless of the effect of this move on the other players. Furthermore, we consider that, whenever a player decides to switch from one coalition to another, the player updates its history set $h(i)$. Hence, given a partition Π, whenever a player i decides to leave coalition $S_m \in \Pi$ to join a different coalition, coalition S_m is automatically stored by player i in its history set $h(i)$.

Consequently, we propose a coalition formation algorithm composed of three main phases: Task discovery, hedonic coalition formation, and data collection. In the first phase, the central command receives information about the existence of tasks that require servicing and informs the agents of the locations and characteristics of the tasks (e.g., the arrival rates). Hence, the agents start by having full knowledge of the initial partition Π_{initial}. Once the agents are aware of the tasks, they broadcast their own presence to the tasks. Consequently, the players can interact with each other, for performing coalition formation. Hence, the second phase of the algorithm is the hedonic coalition formation phase. In this phase, all the players (tasks and agents) investigate the possibility of performing a switch operation. For identifying potential switch operations, given complete knowledge about the network (which can be gathered in different methods as will be discussed in

Subsection IV-C)), each agent investigates its top preference, and decides to perform a switch operation, if possible through (9). As one can easily see through (7), in the proposed model, no coalition composed of tasks-only would ever form since such a coalition would always generate a 0 utility. As a direct result of this property, the tasks are only interested in switching to coalitions that contain at least a single agent. Hence, from a tasks' perspective, for determining its preferred switch operation, each task needs only to negotiate with existing agents in order to enquire on the amount of utility it can obtain by joining with this agent. By doing so, each task can determine the switch operation it is interested in making at a given time. We consider that, the players make sequential switch decisions[7]. For any agent, a switch operation is easily performed as the agent can leave its current coalition and join the new coalition, if (9) is satisfied. For the tasks, any task that finds out a possibility to switch, can autonomously request, over a control channel with the concerned agent, to be added to the coalition of interest (which would always contain at least one agent with whom the task previously negotiated). The convergence of the proposed hedonic coalition formation algorithm during this phase is guaranteed as follows:

Theorem 1: Starting from any initial network partition Π_{initial}, the proposed hedonic coalition formation phase of the proposed algorithm always converges to a final network partition Π_f composed of a number of disjoint coalitions.

Proof: For the purpose of this proof, we denote $\Pi_{n_k}^k$ as the partition formed during the time k when player $i \in \mathcal{N}$ decides to act after n_k switch operations have previously occurred (the index n_k denotes the number of switch operations performed by one or more players up to time k). Given any initial starting partition $\Pi_{\text{initial}} = \Pi_0^1$, the hedonic coalition formation phase of the proposed algorithm consists of a sequence of switch operations. As per Definition 5, every switch operation transforms the current partition Π into another partition Π', hence, the hedonic coalition formation is composed of a sequence of switch operations, yielding the following transformations (as an example)

$$\Pi_0^1 = \Pi_0^2 \to \Pi_1^3 \to \ldots \to \Pi_{n_L}^L \ldots \to \ldots \to \Pi_{n_T}^T, \tag{17}$$

where the operator $\to$ indicate the occurrence of a switch operation. In other words, $\Pi_{n_k}^k \to \Pi_{n_{k+1}}^{k+1}$, implies that during turn k, a certain player i made a single switch operation which yielded a new partition $\Pi_{n_{k+1}}^{k+1}$ at the turn $k+1$. By inspecting the preference relations defined in (9) and (11), it can be seen that every single switch operation leads to a partition that has not yet been visited (new partition). Hence, for any two partitions $\Pi_{n_k}^k$ and $\Pi_{n_l}^l$ in the transformations of (17), such that $n_k \neq n_l$, i.e., $\Pi_{n_l}^l$ is a result of the transformation of

---

[7]This order of switch operations is referred to as the *order of play* hereafter.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON MOBILE COMPUTING

19

$\Pi_{n_k}^k$ (or vice versa) after a number of switch operations $|n_l - n_k|$, we have that $\Pi_{n_k}^k \neq \Pi_{n_l}^l$ for any two turns k and l.

Given this property and the well known fact that the number of partitions of a set is *finite* and given by the Bell number [30], the number of transformations in (17) is finite, and, hence, the sequence in (17) will always terminate and converge to a final partition $\Pi_f = \Pi_{n_T}^T$. Hence, the hedonic coalition formation phase of the proposed algorithm always converges to a final network partition Π_f composed of a number of disjoint coalitions consisting of agents and tasks, which completes the proof. ∎

The stability of the final partition Π_f resulting from the convergence of the proposed algorithm can be studied using the following stability concept from hedonic games [32]:

Definition 6: A partition $\Pi = \{S_1, \ldots, S_l\}$ is *Nash-stable* if $\forall i \in \mathcal{N}$, $S_\Pi(i) \succeq_i S_k \cup \{i\}$ for all $S_k \in \Pi \cup \{\emptyset\}$ (for agents $\succeq_i = \succeq_\mathcal{M}, \forall i \in \mathcal{N} \cap \mathcal{M}$ and for tasks $\succeq_i = \succeq_\mathcal{T}, \forall i \in \mathcal{N} \cap \mathcal{T}$).

In other words, a coalition partition Π is Nash-stable, if no player has an incentive to move from its current coalition to another coalition in Π or to deviate and act alone. Furthermore, a Nash-stable partition Π implies that there does not exist any coalition $S_k \in \mathcal{N}$ such that a player i strictly prefers to be part of S_k over being part of its current coalitions, while all players of S_k do not get hurt by forming $S_k \cup \{i\}$. This is the concept of individual stability, which is formally defined as follows [32]:

Definition 7: A partition $\Pi = \{S_1, \ldots, S_l\}$ is *individually stable* if there do not exist $i \in \mathcal{N}$, and a coalition $S_k \in \Pi \cup \{\emptyset\}$ such that $S_k \cup \{i\} \succ_i S_\Pi(i)$ and $S_k \cup \{i\} \succeq_j S_k$ for all $j \in S_k$ (for agents $\succeq_i = \succeq_\mathcal{M}, \forall i \in \mathcal{N} \cap \mathcal{M}$ and for tasks $\succeq_i = \succeq_\mathcal{T}, \forall i \in \mathcal{N} \cap \mathcal{T}$ for tasks).

As already noted, a Nash-stable partition is individually stable [32]. For the proposed hedonic coalition formation phase of the proposed algorithm, we have the following:

Proposition 2: Any partition Π_f resulting from the hedonic coalition formation phase of the proposed algorithm is Nash-stable, and, hence, individually stable.

Proof: First and foremost, for any partition Π, no player (agent or task) $i \in \mathcal{N}$ has an incentive to leave its current coalition, and act alone as per the utility function in (7). Assume that the partition Π_f resulting from the proposed algorithm is not Nash-stable. Consequently, there exists a player $i \in \mathcal{N}$, and a coalition $S_k \in \Pi_f$ such that $S_k \cup \{i\} \succ_i S_{\Pi_f}(i)$, hence, player i can perform a *switch* operation which contradicts with the fact that Π_f is the result of the convergence of the proposed algorithm (Theorem 1). Consequently, any partition Π_f resulting from the hedonic coalition formation phase of the proposed algorithm is Nash-stable, and, hence, by [32], this resulting partition is also individually stable. ∎

Following the formation of the coalitions and the convergence of the hedonic coalition formation phase to a Nash-stable partition, the last phase of the algorithm entails the actual data collection by the agents.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON MOBILE COMPUTING

20

TABLE I

THE PROPOSED HEDONIC COALITION FORMATION ALGORITHM FOR TASK ALLOCATION IN WIRELESS NETWORKS.

Initial State

The network is partitioned by $\Pi_{\text{initial}} = \{S_1, \ldots, S_k\}$. At the beginning of all time $\Pi_{\text{initial}} = \mathcal{N} = \mathcal{M} \cup \mathcal{T}$

with no tasks being serviced by any agent.

Three Phases for the Proposed Hedonic Coalition Formation Algorithm

Phase I - Task Discovery:

a) The command center is informed by one or multiple owners about the existence and characteristics

of new tasks.

b) The central command center conveys the information on the initial network partition Π_{initial} by entering

the information in appropriate databases such as those used for example, in cognitive radio networks for

primary user information distribution [35], or in unmanned aerial vehicles operation [36], [37].

Phase II - Hedonic Coalition Formation:

In this phase, hedonic coalition formation occurs as follows:

repeat

For every player $i \in \mathcal{N}$, given a current partition Π_{current} (in the first round $\Pi_{\text{current}} = \Pi_{\text{initial}}$).

a) Player i investigates possible switch operations using the preferences given, respectively, by (9)

and (11) for the agents and tasks.

b) Player i performs the switch operation that maximizes its payoff as follows:

b.1) Player i updates its history $h(i)$ by adding coalition $S_{\Pi_{\text{current}}}(i)$, before leaving it.

b.2) Player i leaves its current coalition $S_{\Pi_{\text{current}}}(i)$.

b.3) Player i joins the new coalition that maximizes its payoff.

until convergence to a final Nash-stable partition Π_f.

Phase III - Data Collection

a) The network is partitioned using Π_{final}.

b) The agents in each coalition $S_k \in \Pi_{\text{final}}$ *continuously* perform the following operations, i.e., act as a

polling system with exhaustive strategy and switchover times:

b.1) Visit a first task in their respective coalitions.

b.2) The collector-agents collect the data from the task that is being visited.

b.3) The collector-agents transmit the data using wireless links to the central receiver either directly,

or through other relay-agents.

b.4) Once the queue of the current is empty, visit the next task.

The order in which the tasks are visited is determined by the nearest neighbor solution to the traveling

salesman problem (Section III, Property 3). This third phase is continuously repeated and performed by

all the agents in Π_f for a fixed period of time Ψ (for static environments $\Psi = \infty$).

Adaptation to environmental changes (periodic process)

a) In the presence of environmental changes, such as the deployment of new tasks, the removal of

existing tasks, or periodic low mobility of the tasks, the third phase of the algorithm is performed

continuously only for a *fixed* period of time Ψ.

b) After Ψ elapses, the first two phases are repeated to allow the players to self-organize and adapt the

topology to these environmental changes.

c) This process is repeated periodically for networks where environmental changes may occur.

In this phase, the agents move from one task to the other, in their respective coalitions, collecting the data

and transmitting it to the central receiver, similar to a polling system, as explained in Sections II and III. A

summary of the proposed algorithm is shown in Table I.

The proposed algorithm, as highlighted in Table I, can adapt the network topology to environmental changes such as the deployment of new tasks, the removal of a number of existing tasks, or a periodic low mobility of the tasks (in the case where the tasks represent mobile sensor devices for example). For this purpose, the first two phases of the algorithm shown in Table I are repeated periodically over time, to adapt to any changes that occurred in the environment. With regards to mobility, we only consider the cases where the tasks are mobile for a fixed period of time with a velocity that is smaller than that of the agents η. In the presence of such a mobile environment, the central command center, through Phase I of the algorithm in Table I informs the agents of the new tasks locations (periodically) and, thus, during Phase II of the proposed algorithm, both agents and tasks can react to the environment changes, and modify the existing topology. As per Theorem 1 and Proposition 2, regardless of the starting position, the players will always self-organize into a Nash-stable partition, even after mobility, the deployment of new tasks or the removal of existing tasks. In summary, in a changing environment, the first two phases of the algorithm in Table I are repeated periodically, after a certain fixed period of time Ψ has elapsed during which the players were involved in Phase III and the actual data collection and transmission occurred. Finally, whenever a changing environment is considered, the players are also allowed to periodically clear their history, so as to allow them to explore all the new possibilities that the changes in the environment may have yielded.

C. Distributed Implementation Possibilities

For implementation, first and foremost, in the proposed model, as shown in Figure 1, we clearly distinguish between two inherently different entities: The command center, which is the intelligence that has some control over the agents and the central receiver which is a node in the network that is connected to the command center and which would receive the data transmitted by the agents (this distinction can be, for example, analogous to the distinction between a radio network controller and a base station in cellular networks). In practice, the central command can be, for example, a node that owns a number of agents and controls a large area which is divided into smaller areas with each area represented by the illustration of Figure 1. Hence, each such small area is a region having its own central receiver and where a subset of agents needs to operate and perform coalition formation using our model. In other scenarios, the command center can also be a satellite system that controls groups of agents with each group deployed in a different area (notably when the agents are unmanned aerial vehicles for example). In contrast, the central receiver is simply a wireless node that receives the data from the agents and, subsequently, the command center can obtain this data from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON MOBILE COMPUTING

22

all receivers in its controlled area (e.g., through a backbone network)[8].

For performing coalition formation, the agents and tasks are required to know different information. From the agents perspective, in order to perform a switch operation, each agent is required to obtain data on the location of the tasks (hence, consequently deducing the hop distance D_{ij} between any two tasks i and j) as well as on the arrival rates $\lambda_i, i \in \mathcal{T}$ of these tasks. As a first step, whenever a tasks' owner (e.g., a service provider or a third party) requires that its tasks be serviced, it will give the details and characteristics of these tasks to the network operator (through service-level agreements for example) which would enter these details into the command center. Subsequently, the command center can insert this information into appropriate databases that the agents can access through, for example, an Internet connection. Such a transfer of information through active databases has been recently utilized in many communication architectures, for example, in cognitive radio network for primary user information distribution [35], or in unmanned aerial vehicles operation [36], [37]. In cases where the command center controls only a single set of agents and a single area, this information can be, instead, broadcast directly to the agents. Further, the agents are also required to know the capabilities of each others, notably, the link transmission capacity $\mu_i, \forall i \in \mathcal{M}$ and the velocity (which can be used to deduce the switchover times). As the agents are all owned by a single operator, this information can be easily fed to the agents at the beginning of all time prior to their deployment, and, thus, does not require any additional communication during coalition formation.

From the tasks perspective, the amount of information that needs to be known is much less than that of the agents, notably since the tasks are, in general, resource-limited entities. For instance, as mentioned in the previous section, for performing coalition formation, the tasks do not need to know about the existence or the characteristics of each others. The main information that needs to be known by the tasks is the actual presence of agents. The agents can initially announce/broadcast their presence to the tasks as soon as they enter into the network. Subsequently, the tasks need only to be able to enquire, over a control channel, about the potential utility they would receive from joining the coalition of a particular agent (which can contain other tasks or agents but this is transparent from the perspective of the tasks). The main reason for this is that the tasks have no benefit in forming coalitions that have no agents since such coalitions generate 0 utility for the tasks. Hence, from the point of view of the tasks, they would see every agent as a black box which can provide a certain payoff (communicated over a control channel during negotiation phase), and, based on this, they decide to join the coalition one or another agent (if multiple agents are in the same coalition then

---

[8]This model can easily accommodate the particular case where the command center and the central receiver coincide, e.g., in a small single-area network.

they would offer the same benefit from the tasks perspective). Note that, for coalitions that contain multiple agents, the task needs only to ask *one* agent about their potential utility. In fact, this agent can append, along with the information on the utility, a signal to the tasks about other agents that belong to the same coalition. By doing so, the tasks would no longer need to assess whether to join a coalition by enquiring from other agents that belong to the same coalition, i.e., having redundant information. Hence, by sending this additional information, the agent will enable the tasks to avoid doing multiple processing for the same enquiry.

Consequently, given the information that needs to be known by each player, the proposed algorithm can be implemented in a distributed way since the switch operation can be performed by the tasks or the agents independently of any centralized entity. In this regard, given a partition Π, in order to determine its preferred switch operation, an agent would assess the payoff it would obtain by joining with any coalition in Π, except for singleton coalitions composed of agents only. For the tasks, given Π, each task negotiates with only the agents (and the coalition to which they belong) in the network in order to evaluate its payoff and decide on a switch operation. By adopting a distributed implementation, one would reduce the overhead and computational load on the command center, notably when this command center is controlling numerous areas with different groups of agents (each such area is represented by the model of Figure 1). Further, the distributed approach allows to decentralize the intelligence, and, thus, reduces the detrimental effects on the network and the tasks' owners that can be caused by failures or malicious behavior at the command center level. It is also important to note that the distributed approach complies better with the nature of both the agents and the tasks. On one hand, the agents are inherently autonomous nodes (partially controlled by the command center) that need to operate on their own and, thus, make distributed decisions [1–3], [6–8]. On the other hand, the tasks are independent entities that belong to different owners. Consequently, the tasks are apt to make their own decisions regarding coalition formation and are, generally, unwilling to accept a coalitional structure imposed by an external entity such as the command center. Nonetheless, we note that a centralized approach can also be adopted for the proposed algorithm notably in small networks where, for example, the command center coincides with the central receiver and owns all the tasks.

Regarding complexity, the main complexity lies in the switch operation, the solution to the traveling salesman problem, i.e., determining the order in which the tasks are visited within a coalition in order to evaluate the utility function, and the assignment of agents as either collectors or relays. For instance, given a present coalitional structure Π where each coalition in Π has *at least one task*, for every agent, the computational complexity of finding its next coalition, i.e., performing a switch operation, is easily seen to be $O(|\Pi|)$, and the worst case scenario is when all the tasks act alone, in that case $|\Pi| = T$. In contrast, for the tasks, the worst case complexity is $O(M)$ since, in order to make a switch operation, the tasks need only

to negotiate with agents. With regards to the traveling salesman solution, the complexity of the used nearest neighbor solution is well known to be linear in the number of cities, i.e., tasks [25], hence, for a coalition $S_k \in \Pi$, the complexity of finding the traveling salesman solution is simply $O(|S_k \cap \mathcal{T}|)$, where $S_k \cap \mathcal{T}$ is the set of tasks inside coalition S_k. During coalition formation, i.e., Phase II of the algorithm, whenever a coalition S_k is formed, this coalition needs to compute its own traveling salesman problem, which has a linear complexity $O(|S_k \cap \mathcal{T}|)$, as already mentioned. Certainly, the overall number of traveling salesman problems that should be solved also depends on the number of *new* coalitions that were potentially evaluated for coalition formation prior to convergence to the Nash-stable partition. Hence, the number of traveling salesman solutions that need to be computed is proportional to the number of coalitions (and the identity and number of the tasks within) that negotiated a potential coalition formation *prior* to convergence. This certainly depends on the number of iterations till convergence and the number of switch operations that occurred. Nonetheless, for static environments, *after coalition formation ends*, i.e., in Phase III of the algorithm in Table I, the traveling salesman solution needs to be computed only once *for each coalition* and, afterwards, the network can operate indefinitely (if the environment is static) without any need for the coalitions to re-compute the traveling salesman solution.

Additionally, for determining whether a agent acts as a collector or relay within any coalition, we consider that the players would compute this configuration by inspecting all combinations and selecting the one that maximizes the utility in (7). This computation is done during coalition formation for evaluating the potential utility, and, upon convergence, is maintained during network operation. As the number of agents in a single coalition is generally small, this computation is straightforward, and has reasonable complexity. Finally, in dynamic environments, as the algorithm is repeated periodically and since we consider only periodic low mobility, the complexity of the coalition formation algorithm is comparable to the one in the static environment, but with more runs of the algorithm.

V. SIMULATION RESULTS AND ANALYSIS

For simulations, the following network is set up: A central receiver is placed at the origin of a 4 km $\times$4 km square area with the tasks appearing in the area around it. The path loss parameters are set to $\alpha = 3$ and $\kappa = 1$, the target SNR is set to $\nu_0 = 10$ dB, the pricing factor is set to $\delta = 1$, and the noise variance $\sigma^2 = -120$ dBm. All packets are considered of size 256 bits which is a typical IP packet size. The agents are considered as having a constant velocity of $\eta = 60$ km/h, a transmit power of $\tilde{P} = 100$ mW, and a transmission link capacity of $\mu = 768$ kbps (assumed the same for all agents). Further, we consider two classes of tasks in the network. A first class that can be mapped to voice services having an arrival rate of 32 kbps, and a second class that
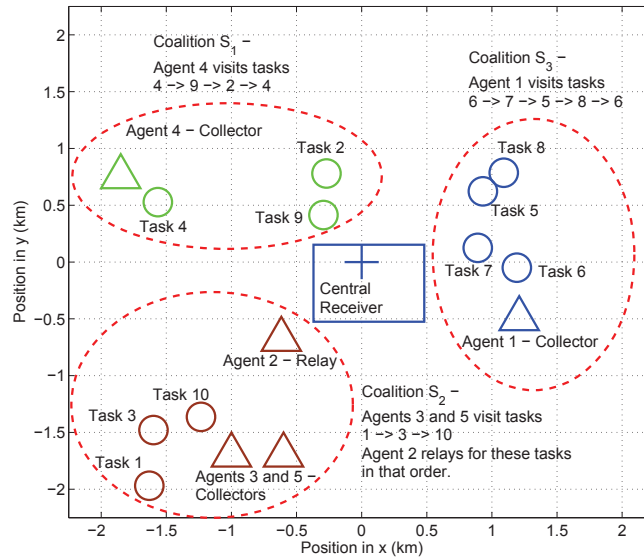
Fig. 2. A snapshot of a final coalition structure resulting from the proposed hedonic coalition formation algorithm for a network of $M = 5$ agents and $T = 10$ tasks. In every coalition, the agents (collectors and relays) visit the tasks continuously in the shown order.

can be mapped to video services, such as the widely known Quarter Common Intermediate Format (QCIF) [38], having an arrival rate 128 kbps. Tasks belonging to each class are generated with equal probability in the simulations. Unless stated otherwise, the throughput-delay tradeoff parameter β is set to 0.7, to indicate services that are reasonably delay tolerant.

In Figure 2, we show a snapshot of the final network partition reached through the proposed hedonic coalition formation algorithm for a network consisting of $M = 5$ agents and $T = 10$ arbitrarily located tasks. In this figure, Tasks 1, 3, and 8 belong to the QCIF video class with an arrival rate of 128 kbps, while the remaining tasks belong to the voice class with an arrival rate of 32 kbps. In Figure 2, we can easily see how the agents and tasks can agree on a partition whereby a number of agents service a group of nearby tasks for data collection and transmission. For the network of Figure 2, the tasks are distributed into three coalitions, two of which (coalitions S_1 and S_3) are served by a single collector-agent. In contrast, coalition S_2 is served by two collectors and one relay. The agents in coalition 2 distributed their roles (relay or collector) depending on the achieved utility. For instance, for coalition S_2, having two collectors and one relay provides a utility of $v(S_2) = 52.25$ while having three collectors yields a utility of $v(S_2) = 10.59$, and having one collector and two relays yields a utility of $v(S_2) = 45.19$. As a result, the case of two collectors and one relay maximizes the utility and is agreed upon between the players. Further, the coalitions in Figure 2 are dynamic, in the sense that, within each coalition, the agents move from one task to the other, collecting and transmitting data to the receiver continuously. The order in which the agents visit the tasks, as indicated in Figure 2, is generated
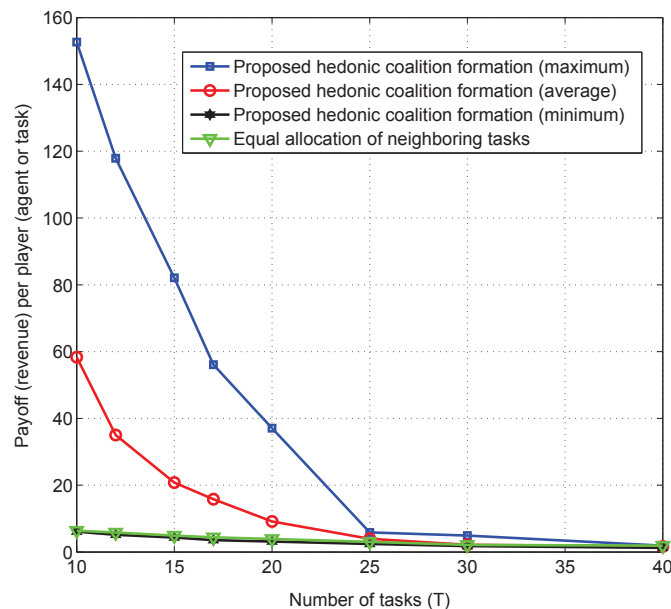
Fig. 3. Performance statistics, in terms of maximum, average and minimum (over the order of play) player payoff (revenue), of the proposed hedonic coalition formation algorithm compared to an algorithm that allocates the neighboring tasks equally among the agents as the number of tasks increases for $M = 5$ agents. All the statistics are also averaged over the random positions of the tasks.

using a nearest neighbor solution for the traveling salesman problem as given by Property 3. For example, consider coalition S_2 in Figure 2. In this coalition, agents 3 an 5 act as a single collector and move from task 1, to task 3, to task 10, and then back to task 1 repeating these visits in a cyclic manner. Concurrently with the collectors movement, agent 2 of coalition S_2, moves and positions itself at the middle of the line connecting the task being serviced by agents 3 and 5 to the central receiver. In other words, when the collectors are servicing task 1 agent 2 is at the middle of the line connecting task 1 to the central receiver, subsequently when the collectors are servicing task 3 agent 2 takes position at the middle of the line connecting task 3 to the central receiver and so on. Finally, note that, for all the coalitions in Figure 2 one can verify that the minimum number of collectors, as per Proposition 1 is approximately 1, (e.g., for coalition S_2, we have $|G_{S_2}|_{\min} = \frac{9}{24}$ thus 1 collector is a minimum), and, hence, this condition is easily satisfied by the coalition formation process.

In Figure 3, we assess the performance of the proposed hedonic coalition formation algorithm, in terms of the payoff (revenue) per player (agent or task) for a network having $M = 5$ agents, as the number of tasks increases. The figure shows the statistics (averaged over the random positions of the tasks), in terms of maximum, average, and minimum over the random order of play. In this figure, we compare the performance with an algorithm that assigns the tasks equally among the agents (i.e., an equal group of neighboring tasks are
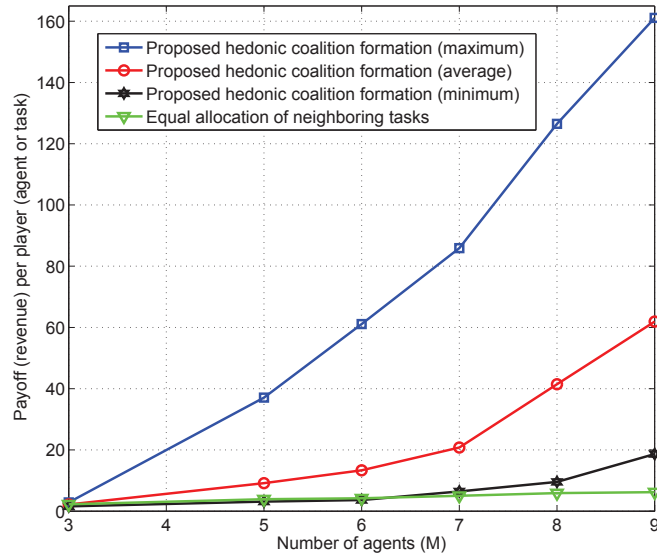
Fig. 4. Performance statistics, in terms of maximum, average and minimum (over the order of play) player payoff (revenue), of the proposed hedonic coalition formation algorithm compared to an algorithm that allocates the neighboring tasks equally among the agents as the number of agents increases for $T = 20$ tasks. All the statistics are also averaged over the random positions of the tasks.

assigned for every agent). Figure 3 shows that the performance of both algorithms is bound to decrease as the number of tasks increases. This is mainly due to the fact that, for networks having a larger number of tasks, the delay incurred per coalition, and, thus, per user increases. This increase in the delay is not only due to the increase in the number of tasks, but also to the increase in the distance that the agents need to travel within their corresponding coalitions (increase in switchover times). In Figure 3, we note that the minimum payoff achieved by the proposed algorithm is comparable to that of the equal allocation. Hence, the performance of the proposed algorithm is clearly lower bounded by the equal allocation algorithm. However, Figure 3 shows that the average and maximum payoff resulting by the proposed algorithm is significantly better than the equal allocation at all network sizes up to $T = 25$ tasks. Albeit this performance improvement decreases with the increase in the number of tasks, the performance, in terms of average payoff per player, yielded by the proposed algorithm is no less than 30.26% better than the equal allocation for up to $T = 25$ tasks. Beyond $T = 25$ tasks, Figure 3 shows that the average and maximum performance of the proposed algorithm is comparable to that of the equal allocation, notably at $T = 40$ tasks. The reduction in the performance gap between the two algorithms for large networks stems from the fact that, as more tasks exist in the network, for a fixed number of agents, the possibility of forming large coalitions, through the proposed hedonic coalition formation algorithm is reduced, and, hence, the structure becomes closer to equal allocation.

In Figure 4, we show the statistics (averaged over the random positions of the tasks), in terms of maximum,

average, and minimum (over the random order of play) payoff per player for a network with $T = 20$ tasks as the number of agents M increases. The performance is once again compared with an algorithm that assigns the tasks equally among the agents (i.e., an equal group of neighboring tasks are assigned for every agent). Figure 4 shows that the performance of both algorithms increases as the number of agents increases. This is mainly due to the fact that when more agents are deployed, the tasks can be better serviced as the delay incurred per coalition decreases and the probability of successful transmission improves. For instance, as more agents enter the network, they can act as either collectors (for improving the delay) or relays (for improving the success probability). We note that, at $M = 3$, the performance statistics of the proposed algorithm converge to the equal allocation algorithm since, for a small number of agents, the flexibility of forming coalitions is quite restricted and equal allocation is the most straightforward coalitional structure. Nonetheless, Figure 4 shows that, as M increases, the performance of the proposed algorithm, in terms of maximum and average payoff achieved, becomes significantly larger than that of the equal allocation algorithm, and this performance advantage increases as more agents are deployed. Finally, Figure 4 also shows that the minimum performance of the proposed algorithm is comparable to the equal allocation algorithm for network with a small number of agents, but as the number of agents increases, the minimum performance of hedonic coalition formation is 29% better than the equal allocation case at $M = 7$, and this advantage increases further with M.

In Figure 5, we show the average and maximum (over the random order of play) coalition size resulting from the proposed algorithm as the number of tasks T increases, for a network of $M = 5$ agents and arbitrarily deployed tasks. These results are averaged over the random positions of the tasks and are compared with the equal allocation algorithm. In this figure, we note that, as the number of tasks increases, the average coalition size for both algorithms increases. For the proposed algorithm, the maximum coalition size also increases with the number of tasks. This is an immediate result of the fact that, as the number of tasks increases, the probability of forming larger coalitions is higher and, hence, our proposed algorithm yields larger coalitions. Further, at all network sizes, the proposed algorithm yields coalitions that are relatively larger than the equal allocation algorithm. This result implies that, by allowing the players (agents and tasks) to selfishly select their coalitions, through the proposed algorithm, the players have an incentive to structure themselves in coalitions with average size lower bounded by the equal allocation. In a nutshell, through hedonic coalition formation, the resulting topology mainly consists of networks composed of a large number of small coalitions as demonstrated by the average coalition size. However, in a limited number of cases, the network topology can also be composed of a small number of large coalitions as highlighted by the maximum coalition size shown in Figure 5.

In Figure 6, we show, over a period of 5 minutes, the frequency in terms of average switch operations per
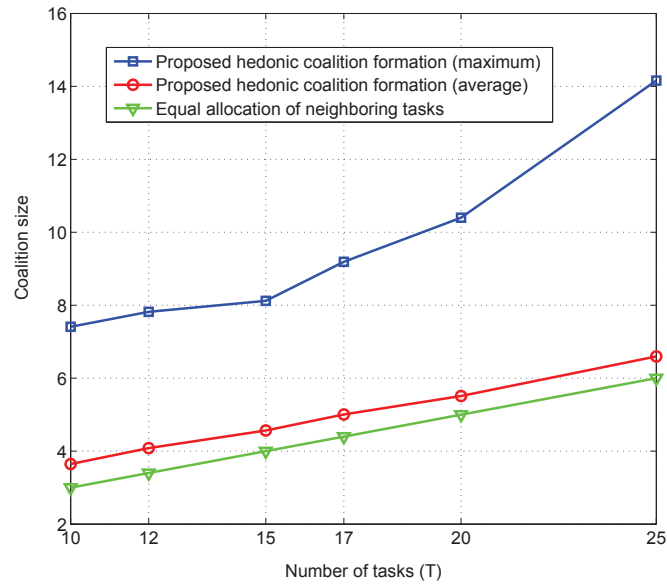
Fig. 5. Average and maximum (over order of play) coalition size yielded by the proposed hedonic coalition formation algorithm and an algorithm that allocates the neighboring tasks equally among the agents, as a function of the number of tasks T for a network of $M = 5$ agents. These statistics are also averaged over random positions of the tasks.
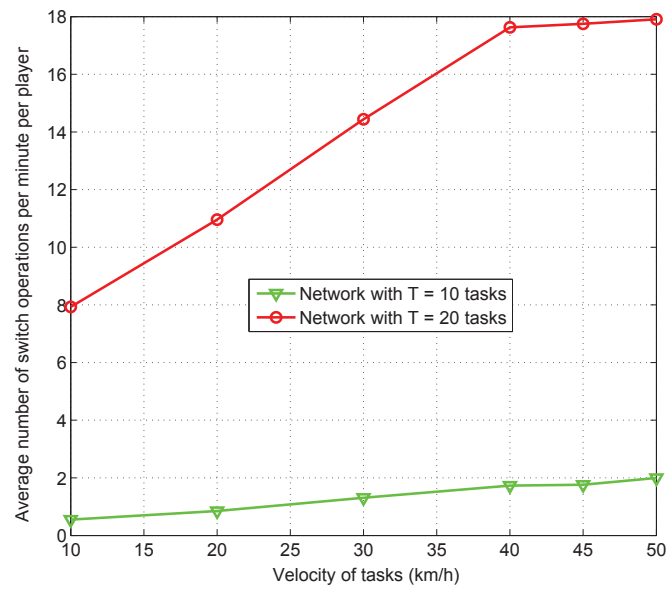

Fig. 6. Frequency of switch operations per minute per player (agent or task) achieved over a period of 5 minutes for different tasks' velocities in wireless network having $M = 5$ agents and different number of mobile tasks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
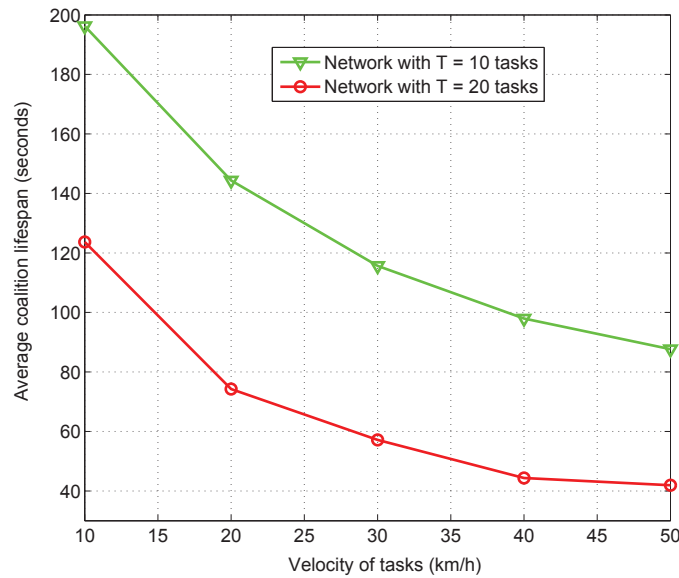
IEEE TRANSACTIONS ON MOBILE COMPUTING

30

Fig. 7. Average coalition lifespan (in seconds) achieved over a period of 5 minutes for different tasks' velocities in wireless network having $M = 5$ agents and different number of mobile tasks. The coalition lifespan indicates the time during which a coalition is present in the network before accepting new partners or breaking into smaller coalitions (due to switch operations).

minute per player (agent or task) achieved for various velocities of the tasks in a mobile wireless network with $M = 5$ agents and different number of tasks. As the velocity of the tasks increases, the frequency of the switch operations increases for both $T = 10$ and $T = 20$ due to the changes in the positions of the various tasks incurred by mobility. Figure 6 shows that the case of $T = 20$ tasks yields a frequency of switch operations significantly higher than the case of $T = 10$ tasks. This result is interpreted by the fact that, as the number of tasks increases, the possibility of finding new partners as the tasks move increases significantly, hence yielding an increase in the topology variation as reflected by the number of switch operations. In summary, this figure shows that the proposed hedonic coalition formation algorithm allows the agents and the tasks to self-organize and adapt their topology to mobility, through adequate switch operations.

The network's adaptation to mobility is further assessed in Figure 7, where we show, over a period of 5 minutes, the average coalition lifespan (in seconds) achieved for various velocities of the tasks in a mobile wireless network with $M = 5$ agents and different number of tasks. The coalition lifespan is defined as the time (in seconds) during which a coalition is present in the mobile network prior to accepting new members or breaking into smaller coalitions (due to switch operations). In this figure, we can see that, as the velocity of the tasks increases, the average lifespan of a coalition decreases. This is due to the fact that, as mobility increases, the possibility of forming new coalitions or splitting existing coalitions increases significantly. For example, for $T = 20$, the coalition lifespan drops from around 124 seconds for a tasks' velocity of 10 km/h
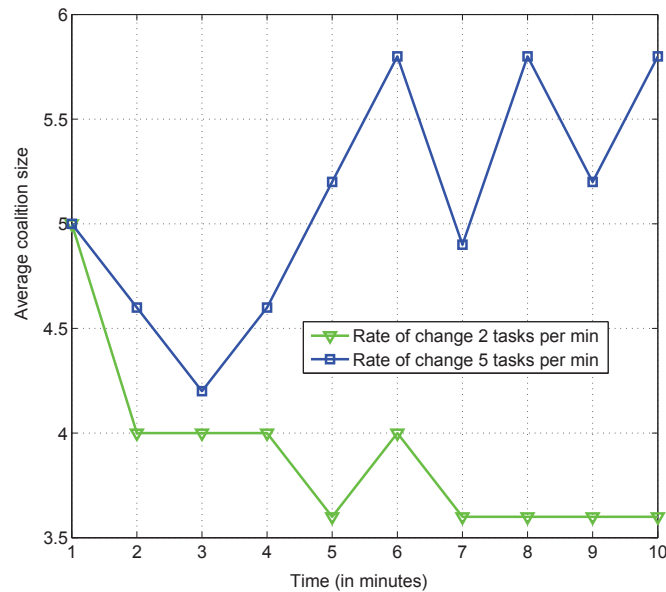
Fig. 8. Topology variation over time as new tasks join the network and existing tasks leave the network with different rates of tasks arrival/departure for a network starting with $T = 15$ tasks and having $M = 5$ agents.

to just under a minute as of 30 km/h, and down to around 42 seconds at 50 km/h. Furthermore, Figure 7 shows that as more tasks are present in the network, the coalition lifespan decreases. For instance, for any given velocity, the lifespan of a coalition for a network with $T = 10$ tasks is significantly larger than that of a coalition in a network with $T = 20$ tasks. This is a direct result of the fact that, for a given tasks' velocity, as more tasks are present in the network, the players are able to find more partners to join with, and hence the lifespan of the coalitions becomes shorter. In brief, Figure 7 provides an interesting assessment of the topology adaptation aspect of the proposed algorithm through the process of forming new coalitions or breaking existing coalitions.

Moreover, for further analysis of the self-adapting aspect of the proposed hedonic coalition formation algorithm, we study the variations of the coalitional structure over time for a network where tasks are entering and leaving the network. For this purpose, in Figure 8, we show the variations of the average (over the random positions of the tasks) number of players per coalition, i.e., the average coalition size, over a period of 10 minutes, as new tasks join the network and/or existing tasks leave the network. The considered network in Figure 8 possesses $M = 5$ agents and starts with $T = 15$ tasks. The results are shown for different rates of change which is defined as the number of tasks that have either entered the network or left the network per minute. For example, a rate of change of 2 tasks per minute indicates that either 2 tasks enter the network every minute, 2 tasks leave the network every minute, or 1 tasks enters the network and another tasks leaves

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

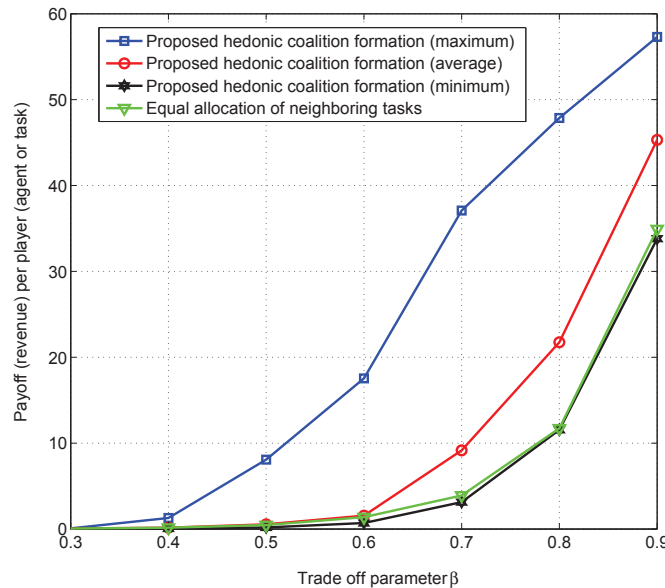IEEE TRANSACTIONS ON MOBILE COMPUTING

32

Fig. 9. Performance statistics, in terms of maximum, average and minimum (over the order of play) player payoff (revenue), of the proposed hedonic coalition formation algorithm compared to an algorithm that allocates the neighboring tasks equally among the agents as the throughput-delay tradeoff parameter β increases for $M = 5$ agents and $T = 20$ tasks. All the statistics are also averaged over the random positions of the tasks.

the network every minute (these cases may occur with equal probability). In this figure, we can see that, as time evolves, the structure of the network is changing, with new coalitions forming and other breaking as reflected by the change in coalitions size. Furthermore, we note that, as the rate of change increases, the changes in the topology increase. For instance, it is seen in Figure 8 that for a rate of change of 5 tasks per minute, the variations in the coalition size are much larger than for the case of 2 tasks per minute (which is almost constant for many periods of time). In summary, Figure 8 shows the network topology variations as tasks enter or leave the network. Note that, after the 10 minutes have elapsed, the network re-enters in the Phase III of the algorithm where data collection and transmission occurs.

In Figure 9, we assess the performance of the proposed hedonic coalition formation algorithm, in terms of the payoff (revenue) per player (agent or task) for a network having $M = 5$ agents and $T = 20$ tasks, as the throughput-delay tradeoff parameter β increases. The figure shows the statistics, in terms of maximum, average, and minimum over the random order of play between the players. In this figure, we can see that, for small β, the performance of the proposed algorithm is comparable to the equal allocation algorithm and the payoffs are generally small. This result is due to the fact that, for small β, the tasks are highly delay sensitive, and the delay component of the utility governs the performance. Hence, for such tasks, the proposed algorithm yields a performance similar to equal allocation. However, as the tradeoff parameter β increases, the maximum and

average utility yielded by our proposed algorithm outperforms the equal allocation algorithm significantly. For instance, as of $\beta = 0.5$, hedonic coalition formation is highly desirable, and presents a performance improvement in terms of average payoff of around 19.56% relative to the equal allocation algorithm (at $\beta = 0.55$, the proposed algorithm has an average payoff of 0.55 while equal allocation has an average payoff of 0.46). This advantage increases with β. Note that, for all tradeoff parameters, the performance of the proposed algorithm, in terms of minimum (over order of play) payoff gained by a player is lower bounded by the equal allocation algorithm and, in average, outperforms the equal allocation algorithm.

VI. CONCLUSIONS

In this paper, we introduced a novel model for task allocation among a number of autonomous agents in a wireless communication network. In the introduced model, a number of wireless agents are required to service several tasks, arbitrarily located in a given area. Each task represents a queue of packets that require collection and wireless transmission to a centralized receiver by the agents. The task allocation problem is modeled as a hedonic coalition formation game between the agents and the tasks that interact in order to form disjoint coalitions. Each formed coalition is mapped to a polling system which consists of a number of agents continuously collecting packets from a number of tasks. Within a coalition, the agents can act either as collectors that move between the different tasks present in the coalition for collecting the packet data, or relays for improving the wireless transmission of the data packets. For forming the coalitions, we introduce an algorithm that allows the players (tasks or agents) to join or leave the coalitions based on their preferences which capture the tradeoff between the effective throughput and the delay achieved by the coalition. We study the properties and characteristics of the proposed model, we show that the proposed hedonic coalition formation algorithm always converges to a Nash-stable partition, and we study how the proposed algorithm allows the agents and tasks to take distributed decisions for adapting the network topology to environmental changes such as the deployment of new tasks, the removal of existing tasks or the mobility of the tasks. Simulation results show how the proposed algorithm allows the agents and tasks to self-organize into independent coalitions, while improving the performance, in terms of average player (agent or task) payoff, of at least 30.26% (for a network of 5 agents with up to 25 tasks) relatively to a scheme that allocates nearby tasks equally among the agents. In a nutshell, by combining concepts from wireless networks, queueing theory and novel concepts from coalitional game theory, we proposed a new model for task allocation among autonomous agents in communication networks which is well suited for many practical applications such as data collection, data transmission, autonomous relaying, operation of message ferry (mobile base stations), surveillance, monitoring, or maintenance in next generation wireless networks.

REFERENCES

[1] M. Debbah, "Mobile flexible networks: The challenges ahead," in *Proc. International Conference on Advanced Technologies for Communications*, Hanoi, Vietnam, Oct. 2008.

[2] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Başar, "A game-based self-organizing uplink tree for VoIP services in IEEE 802.16j networks," in *Proc. Int. Conf. on Communications*, Dresden, Germany, Jun. 2009.

[3] D. Niyato, E. Hossein, and Z. Han, *Dynamic spectrum access in cognitive radio networks*. Cambridge, UK: Cambridge University Press, 2009.

[4] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot, "Measurement-based self organization of interfering 802.11 wireless access networks," in *Proc. IEEE Conf. on Computer Communications (INFOCOM)*, Alaska, USA, May 2007.

[5] A. Akella, G. Judd, P. Steenkiste, and S.Seshan, "Self management in chaotic wireless deployments," in *Proc. Int. Conf. on Mobile Computing and Networking (MobiCom)*, Cologne, Germany, Sep. 2005.

[6] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Connectivity in vehicular ad hoc networks in presence of wireless mobile base-stations," in *Proc. 7th International Conference on ITS Telecommunications*, Sophia Antipolis, France, Jun. 2007.

[7] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proc. of ACM MobiHoc*, Florence, Italy, May 2006.

[8] Y. Shi and Y. T. Hou, "Theoretical results on base station movement problem for sensor network," in *Proc. IEEE Conf. on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, Apr. 2008.

[9] B. Gerkey and M. J. Mataric, "A formal framework for the study of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, Sep. 2004.

[10] M. Alighanbari and J. How, "Robust decentralized task assignment for cooperative UAVs," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, Colorado, USA, Aug. 2006.

[11] D. M. Stipanovic, P. F. Hokayem, M. W. Spong, and D. D. Siljak, "Cooperative avoidance control for multi-agent systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 699–706, Sep. 2007.

[12] Q. Chen, M. Hsu, U. Dayal, and M. Griss, "Multi-agent cooperation, dynamic workflow and XML for e-commerce automation," in *Proc. of Int. Conf. on Autonomous agents*, Catalonia, Spain, Jun. 2000.

[13] O. Shehory and S. Kraus, "Methods for task allocation via agent coalition formation," *Artifical Intelligence Journal*, vol. 101, pp. 165–200, May 1998.

[14] R. Beard, D. Kingston, M.Quigley, D. Snyder, R. Christiansen, W. Johnson, T. Mclain, and M. Goodrich, "autonomous vehicle technologies for small fixed wing UAVs," *AIAA Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 1, pp. 92–108, January 2005.

[15] Z. Han, A. Swindlehurst, and K. J. Liu, "Optimization of MANET connectivity via smart deployment/movement of unmanned air vehicles," *IEEE Trans. Vehicular Technology*, to appear, 2009.

[16] D. L. Gu, G. Pei, H. Ly, M. Gerla, B. Zhang, and X. Hong, "UAV aided intelligent routing for ad hoc wireless network in single area theater," in *Proc. IEEE Wireless Communications and Networking Conf.*, Chicago, IL, Sep. 2000, pp. 1220–1225.

[17] K. Xu, X. Hong, M. Gerla, H. Ly, and D. L. Gu, "landmark routing in large wireless battlefield networks using UAVs," in *in Proc. of IEEE Military Communication Conference*, Washington DC, Oct. 2001, pp. 230–234.

[18] D. L. Gu, H. Ly, X. Hong, M. Gerla, G. Pei, and Y. Lee, "C-ICAMA, a centralized intelligent channel assigned multiple access for multi-layer ad-hoc wireless networks with UAVs," in *Proc. IEEE Wireless Communications and Networking Conf.*, Chicago, IL, Sep. 2000, pp. 879–884.

[19] J. Proakis, *Digital Communications*. New York, USA: 4th ed., McGraw-Hill, 2001.

[20] R. B. Myerson, *Game Theory, Analysis of Conflict*. Cambridge, MA, USA: Harvard University Press, Sep. 1991.

[21] H. Takagi, *Analysis of Polling Systems*. Cambridge, MA, USA: The MIT Press, Apr. 1986.

[22] H. Levy and M. Sidi, "Polling systems: applications, modeling, and optimization," *IEEE Trans. Commun.*, vol. 10, pp. 1750–1760, Oct. 1990.

[23] Y. Li, H. S. Panwar, and J. Shao, "Performance analysis of a dual round robin matching switch with exhaustive service," in *Proc. IEEE Global Communication Conference*, Taipei, Taiwan, Nov. 2002.

[24] V. Vishnevsky and O. Semenova, "The power-series algorithm for two-queue polling system with impatient customers," in *in Proc. of Int. Conf. on Telecom.*, Stockholm, Sweden, Jun. 2008.

[25] D. Applegate, R. M. Bixby, V. Chvatal, and W. J. Cook, *The traveling salesman problem: a computational study*. Princeton, NJ, USA: Princeton University Press, 2006.

[26] L. Kleinrock, "Power and deterministic rules of thumb for probabilistic problems in computer communications," in *Proc. Int. Conf. on Communications*, Boston, USA, Jun. 1979.

[27] W. K. Ching, "A note on the convegence of asynchronous greedy algorithm with relaxation in a multiclass queueing system," *IEEE Commun. Lett.*, vol. 3, pp. 34–36, Feb. 1999.

[28] E. Altman, T. Başar, and R. Srikant, "Nash equilibria for combined flow control and routing in networks: asymptotic behaviour for a large number of users," *IEEE Trans. Automtatic Control*, vol. 47, pp. 917–930, Jun. 2002.

[29] V. Vukadinovic and G. Karlsson, "Video streaming in 3.5G: On throughput-delay performance of proportional fair scheduling," in *Proc. Int. Symp. on Modeling, Analysis and Simulation of Comp. and Telecom. Systems*, California, USA, Sep. 2006.

[30] D. Ray, *A Game-Theoretic Perspective on Coalition Formation*. New York, USA: Oxford University Press, Jan. 2007.

[31] G. Demange and M. Wooders, *Group Formation in Economics*. New York, USA: Cambridge University Press, 2006.

[32] A. Bogomonlaia and M. Jackson, "The stability of hedonic coalition structures," *Games and Economic Behavior*, vol. 38, pp. 201–230, Jan. 2002.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON MOBILE COMPUTING

35

[33] E. Diamantoudi and L. Xue, "Farsighted stability in hedonic games," *Social Choice Welfare*, vol. 21, pp. 39–61, Jan. 2003.

[34] J. Drèze and J. Greenberg, "Hedonic coalitions: Optimality and stability," *Econometrica*, vol. 48, pp. 987–1003, Jan. 1980.

[35] S. Shellhammer, A. Sadek, and W. Zhang, "Technical challenges for cognitive radio in TV white space spectrum," in *Proc. of Information Theory and Applications Workshop (ITA)*, San Diego, USA, Feb. 2009.

[36] G. Calhoun, M. H. Drape, M. F. Abernathy, M. Patzek, and F. Delgado, "Synthetic vision system for improving unmanned aerial vehicle operator situation awareness," in *Proc. of the International Society for Optical Engineering*, Orlando, FL, USA, Mar. 2005.

[37] M. Brohede and S. F. Andler, "Distributed simulation communication through an active real-time database," in *Proc. of NASA Goddard/IEEE Software Engineering Workshop*, Greenbelt, MD, USA, Dec. 2002.

[38] L. Wainfan, *Challenges in Virtual Collaboration: Videoconferencing Audioconferencing and Computer-Mediated Communications*. California, USA: RAND Corporation, Jul. 2005.

[39] W. Saad, Z. Han, T. Başar, M. Debbah, and A. Hjørungnes, "A selfish approach to coalition formation among unmanned aerial vehicles in wireless networks," in *Proc. of Int. Conference on Game Theory for Networks (GameNets)*, Istanbul, Turkey, May 2009.
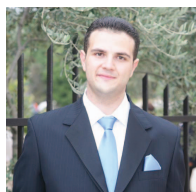
Walid Saad received his B.E. degree in Computer and Communications Engineering from the Lebanese University, Faculty of Engineering II, in 2004, his M.E. in Computer and Communications Engineering from the American University of Beirut (AUB) in 2007, and his Ph.D degree from the University of Oslo in 2010.

From August 2008 till July 2009 he was a visiting scholar in the Coordinated Science Laboratory at the University of Illinois at Urbana Champaign. As of Fall 2010, he will start working as a Postdoctoral Research Associate at the Electrical and Computer Engineering Department at Princeton University.

His research interests include applications of game theory in wireless networks, coalitional game theory, cognitive radio, autonomous communication and wireless communication systems (UMTS, WiMAX, LTE, etc). He was the first author of the papers that received the Best Paper Award at the 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), in June 2009 and at the 5th International Conference on Internet Monitoring and Protection (ICIMP) in May 2010.

Zhu Han (S'01-M'04-S'09) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor in Boise State University, Idaho. Currently, he is an Assistant Professor in Electrical and Computer Engineering Department at University of Houston, Texas. In June-August 2006, he was a visiting scholar in Princeton University. In May-August 2007, he was a visiting professor in Stanford University. In May-August 2008, he was a visiting professor in University of Oslo, Norway and Supelec, Paris, France. In July 2009, he was a visiting professor in the University of Illinois at Urbana-Champion. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, wireless multimedia, and security.

Dr. Han is an NSF CAREER award recipient 2010. Dr. Han is an Associate Editor of IEEE Transactions on Wireless Communications since 2010. Dr. Han was the MAC Symposium vice chair of IEEE Wireless Communications and Networking Conference, 2008. Dr. Han was the Guest Editor for Special Issue on Cooperative Networking Challenges and Applications (IEEE Journal on Selected Areas in Communications) Fairness of Radio Resource Management Techniques in Wireless Networks (EURASIP Journal on Wireless Communications and Networking), and Special Issue on Game Theory (EURASIP Journal on Advances in Signal Processing). Dr. Han is the coauthor for the papers that won the best paper awards in IEEE International Conference on Communications 2009 and 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt09).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON MOBILE COMPUTING

Tamer Başar (S'71-M'73-SM'79-F'83) received the BSEE degree from Robert College, Istanbul, and the MS, MPhil, and PhD degrees in engineering and applied science from Yale University. After holding positions at Harvard University and Marmara Research Institute (Gebze, Turkey), he joined the University of Illinois at Urbana-Champaign (UIUC) in 1981, where he currently holds the positions of Swanlund Endowed Chair, Center for Advanced Study Professor of Electrical and Computer Engineering, Research Professor at the Coordinated Science Laboratory, and Research Professor at the Information Trust Institute. He is also the Interim Director of the Beckman Institute for Advanced Science and Technology. He has published extensively in systems, control, communications, and dynamic games, and has current research interests in modeling and control of communication networks; control over heterogeneous networks; resource allocation, management and pricing in networks; game-theoretic tools for networks; mobile computing; security issues in computer networks; and robust identification, estimation and control.

Dr. Başar is the Editor-in-Chief of *Automatica* and the *IEEE Expert Now*, Editor of the Birkhäuser Series on *Systems & Control*, Managing Editor of the *Annals of the International Society of Dynamic Games* (ISDG), and member of editorial and advisory boards of several international journals in control, wireless networks, and applied mathematics. He has received several awards and recognitions over the years, among which are the Medal of Science of Turkey (1993); Distinguished Member Award (1993), Axelby Outstanding Paper Award (1995), and Bode Lecture Prize (2004) of the IEEE Control Systems Society (CSS); Millennium Medal of IEEE (2000); Tau Beta Pi Drucker Eminent Faculty Award of UIUC (2004); the Outstanding Service Award (2005) and the Giorgio Quazza Medal (2005) of the International Federation of Automatic Control (IFAC); and the Richard E. Bellman Control Heritage Award (2006) of the American Automatic Control Council (AACC). He is a member of the National Academy of Engineering (of USA), a member of the European Academy of Sciences, a Fellow of IEEE, a Fellow of IFAC, President-Elect of AACC, a past president of CSS, and the founding president of ISDG.

Mérouane Debbah was born in Madrid, Spain. He entered the Ecole Normale Supérieure de Cachan (France) in 1996 where he received his M.Sc and Ph.D. degrees respectively in 1999 and 2002. From 1999 to 2002, he worked for Motorola Labs on Wireless Local Area Networks and prospective fourth generation systems. From 2002 until 2003, he was appointed Senior Researcher at the Vienna Research Center for Telecommunications (FTW) (Vienna, Austria) working on MIMO wireless channel modeling issues. From 2003 until 2007, he joined the Mobile Communications department of the Institut Eurecom (Sophia Antipolis, France) as an Assistant Professor. He is presently a Professor at Supelec (Gif-sur-Yvette, France), holder of the Alcatel-Lucent Chair on Flexible Radio. His research interests are in information theory, signal processing and wireless communications. Mrouane Debbah is the recipient of the 2007 General Symposium IEEE GLOBECOM best paper award as well as the Valuetools 2007 and Valuetools 2008 best student paper awards. He is a WWRF fellow.

Are Hjørungnes (IEEE Senior Member) works as a Professor at the Faculty of Mathematics and Natural Sciences at the University of Oslo, Norway with office located at UNIK - University Graduate Center. He obtained his Sivilingenir (M.Sc.) degree (with honors) in 1995 from the Department of Telecommunications at the Norwegian Institute of Technology in Trondheim, Norway, and his Doktor ingenir (Ph.D.) degree in 2000 from the Signal Processing Group at the Norwegian University of Science and Technology. His current main research areas are signal processing, communications, and wireless networks. From August 2000 to December 2000, he worked as a researcher at Tampere University of Technology, in Finland, within the Tampere International Center for Signal Processing. From March 2001 to July 2002, he worked as a postdoctoral fellow at the Federal University of Rio de Janeiro in Brazil, within the Signal Processing Laboratory. From September 2002 to August 2003, he worked as a postdoctoral fellow at the Helsinki University of Technology in Finland, within the Signal Processing Laboratory. From September 2003 to August 2004, he was working as a postdoctoral fellow at the University of Oslo in Norway, at the Department of Informatics, within the Digital Signal Processing and Image Analysis Group.

He has held visiting appointments at the Image and Signal Processing Laboratory at the University of California, Santa Barbara, the Signal Processing Laboratory of the Federal University of Rio de Janeiro, the Mobile Communications Department at Eurecom Institute in France, the University of Manitoba in Canada, the Alcatel-Lucent Chair at SUPLEC in France, the Department of Electrical and Computer Engineering at the University of Houston in USA, and the Electrical and Computer Engineering Department at University of California, San Diego, USA.

Since March 2007, he has been serving as an Editor for IEEE Transactions on Wireless Communications. In June 2010, he was a Guest Editor of the IEEE Journal of Selected Topics in Signal Processing, in the special issue on "Model Order Selection in Signal Processing Systems". He co-authored the papers winning the best paper awards at IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2007), 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt 2009), and 5th International Conference on Internet Monitoring and Protection, ICIMP 2010.