Hedonic Coalition Formation for Distributed Task Allocation among Wireless Agents

Walid Saad, Zhu Han, Tamer Basar, Me´rouane Debbah, and Are Hjørungnes.

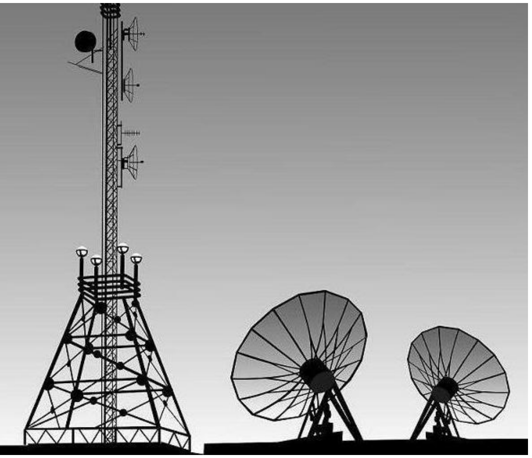
IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 10, NO. 9, SEPTEMBER 2011.

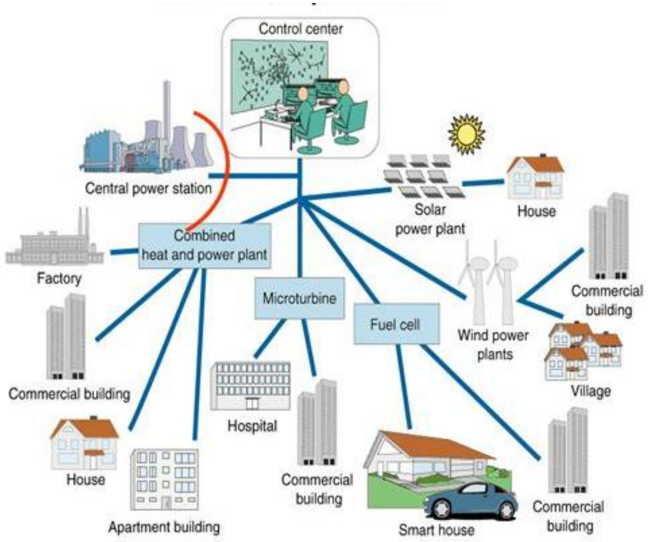
Outline

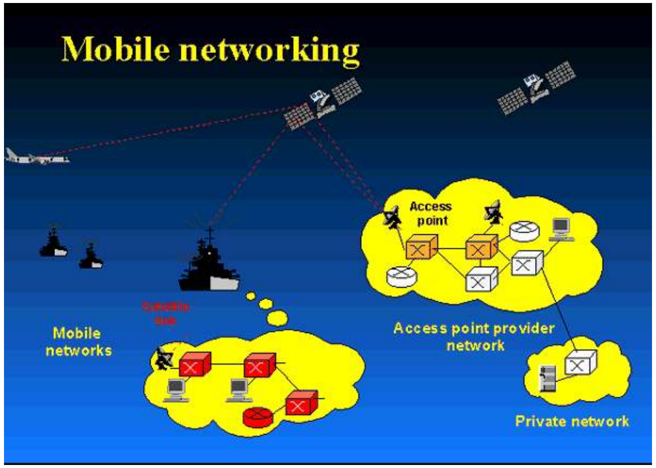
- Background & Motivation
- Introduction
- System Model
- Coalitional Game Formation
  - **Game Formation**
  - **Utility Function**
- Task Allocation as a Hedonic Coalition Formation Game
  - **Hedonic Coalition Formation: Concepts & Model**
  - **Hedonic Coalition Formation: Algorithm**
  - **Distributed Implementation Possibilities**
- Simulation Results & Analysis
- Conclusions

Background

- Communication systems
  - **Large-Scale**
  - **Distributed**
  - **Heterogeneous**

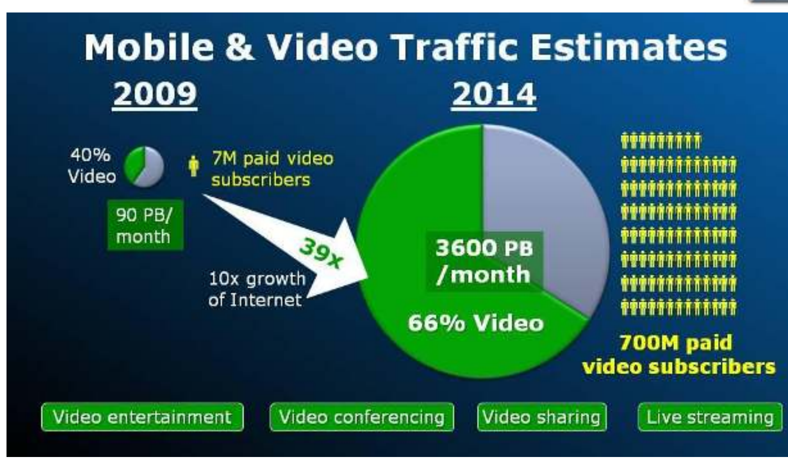






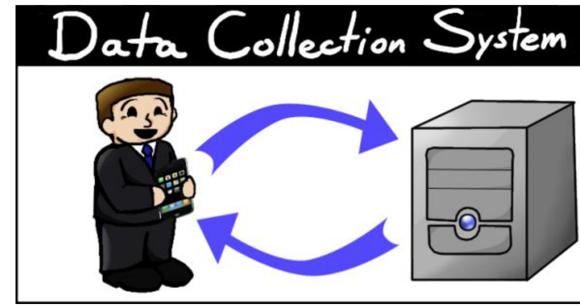
Background

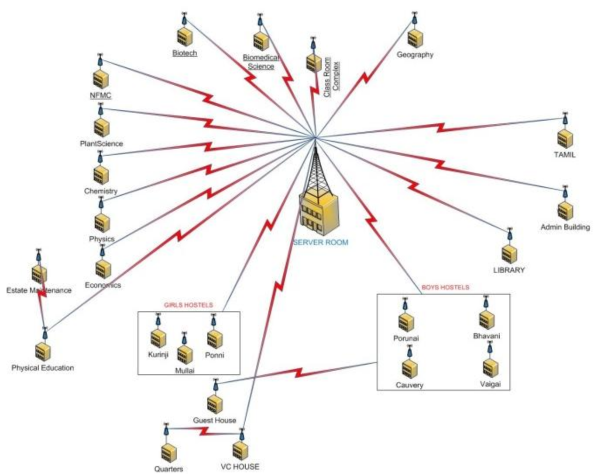
- Challenges
  - **Increase in size, traffic, applications, services**
  - **Need for dynamically**
    - optimizing their performance
    - monitoring their operation
    - reconfiguring their topology

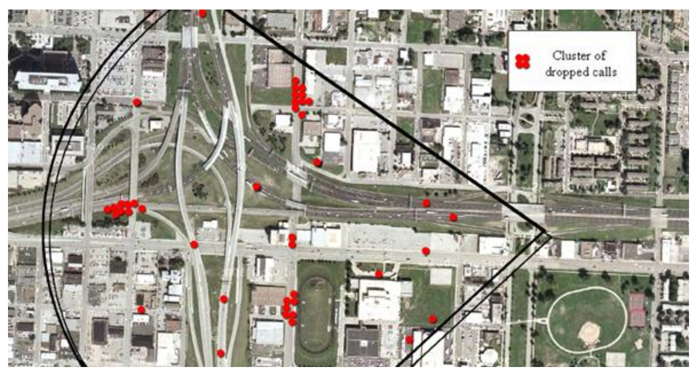


Background

- <u>Self-organizing autonomous nodes </u>serving different level networks
  - **Data collection**
  - **Monitoring**
  - **Optimization**
  - **Management**
  - **…**

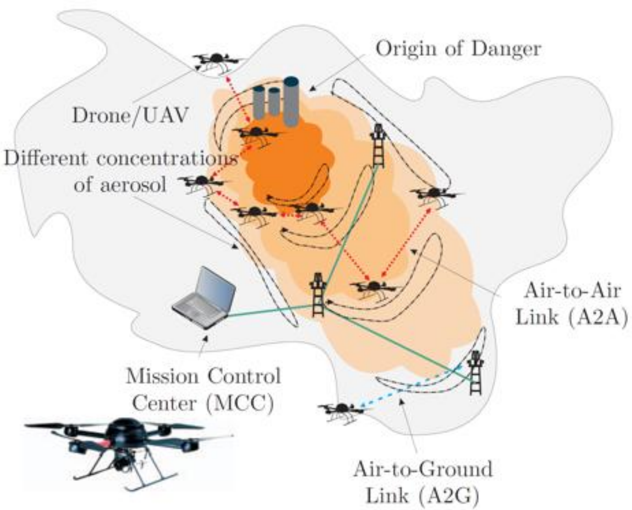


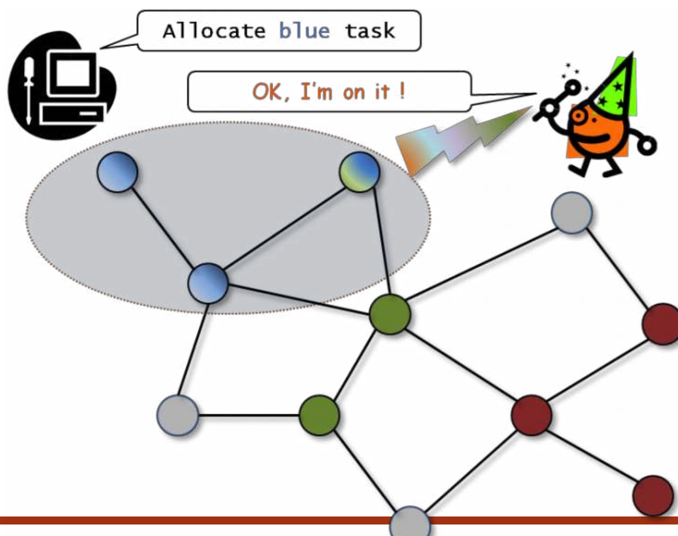




Motivation

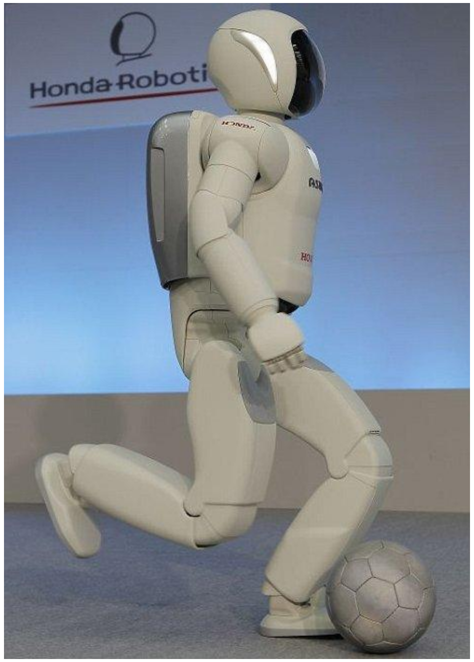
- Next-Generation Networks
  - **Cognitive devices**
  - **Unmanned aerial vehicles**
    - Require the nodes are **autonomous** and **self-adapting**
- Key Problem
  - **Task Allocation among a group of agents**





Introduction

- Applications of Autonomous and Self-adapting agents
  - **Robotics control**
  - **Software systems**



Introduction

- These existing models are unsuitable for task allocation problems due to various reasons
  - **Existing papers are mainly tailored for military operations, computer systems, or software engineering**
  - **The tasks are generally considered as static abstract entities with very simple characteristics and no intelligence**
  - **The existing models do not consider any aspects of wireless communication networks**
    - Characteristics of wireless channel
    - The presence of data traffic
    - The need for wireless data transmission

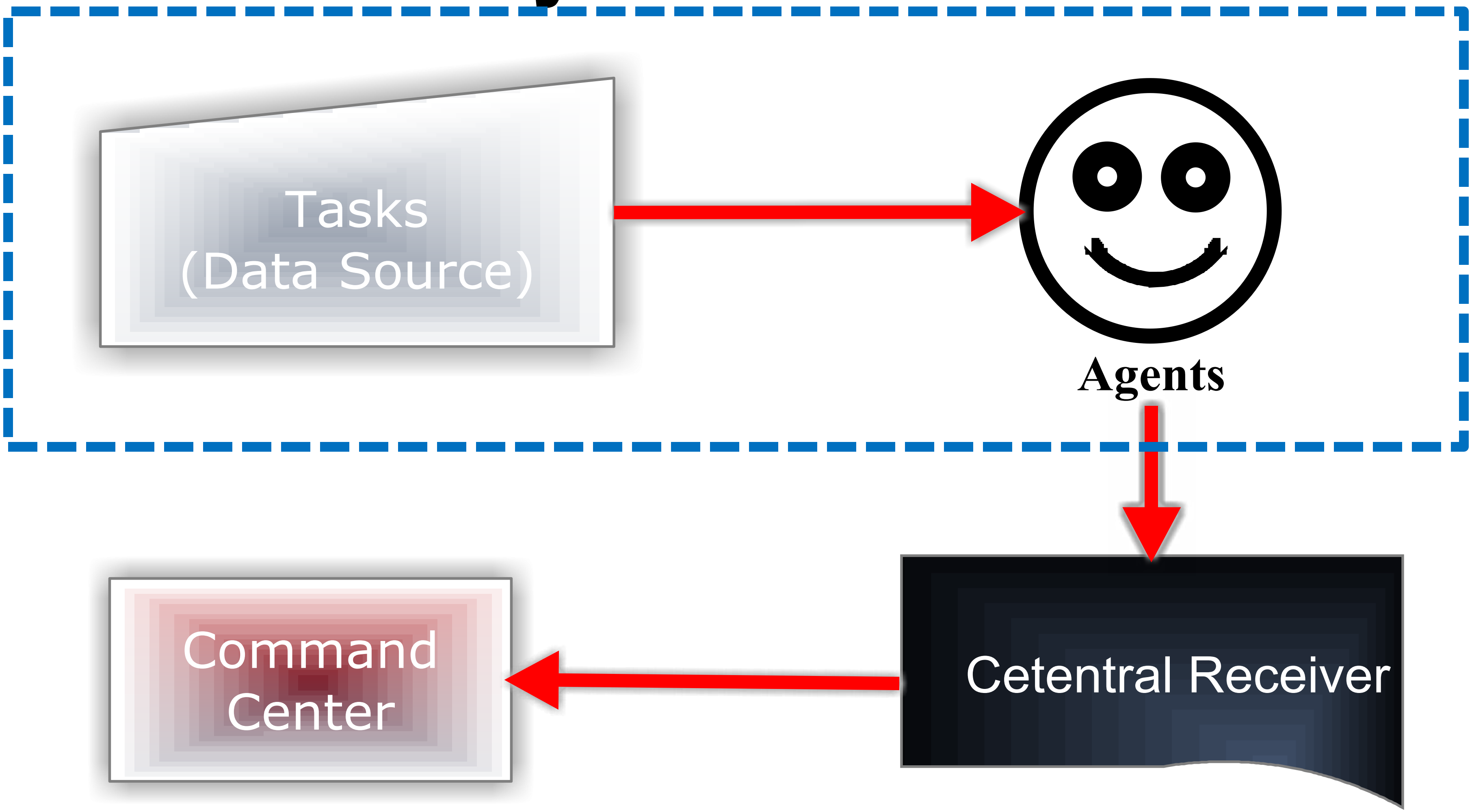
Introduction

- The *main existing contributions* within wireless networking in this area
  - **Deploy unmanned aerial vehicles (UAVs)**
    - Efficiently perform *preassigned* and *predetermined* tasks in numerous applications
      - **Connectivity improvement in ad hoc network**
    - Focus on centralized solutions
      - **Find the optimal locations for the deployment of UAVs**

Introduction

- This paper
  - **Propose a wireless communication-oriented model for the problem of task allocation among a number of autonomous agents**
  - **Address the issue**
    - Task allocation
  - **Environment**
    - Wireless communication systems consisting of autonomous agents
  - **Distributed system**

System Model



System Model

- System representation
  - M **wireless agents**
    - $\mathcal{M} = \{1, 2, \ldots, M\}$
  - **Single network operator**
  - **Central command center**
  - **These agents are required to serve T tasks**
    - $\mathcal{T} = \{1, 2, \ldots, T\}$
    - where $T > M$

Tasks

- Source of Data
- Each task $i \in \mathcal{T}$ represents an M/D/1 queuing system
  - **whereby packets of constant size of B are generated using a Poisson arrival with an average arrival rate of λ_i**
- Consider different classes of tasks
  - **Can represent a group of mobile devices**
    - such as sensors, video surveillance devices, etc.
    - These devices need to buffer their data locally and await to be serviced

Agents

- Service tasks
  - **Move to a task location**
  - **Collect data**
  - **Transmit data using a wireless link to the central receiver**
- Agent $i \in \mathcal{M}$ offers transmission capacity μ_i, in packets/second
  - **Service time:** $\frac{1}{\mu_i}$
- Collector: collecting data
- Relay: transmit data

Total Transmission Capacity

- Link transmission capacity depends solely on the capacities of the agents.
- A group of agents $G \subseteq \mathcal{M}$ are agents for any task, then the total link transmission capacity with which tasks can be serviced by G can be given by

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

Successful Transmission Probability

- *Relays* locate themselves at equal distances from the task to form multi-hop agents.
- 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, $Q_i = \{i_1, i_2, \ldots, i_m\}$, $i_1 = i$ is the task being serviced, i_m is the central server, any other $i_h \in Q_i$ is *relay-agent*.
  - $\Pr_i^{CR} = \prod_{h=1}^{m-1} \Pr_{i_h, i_{h+1}}^{B}$ **ddd**
  - $\Pr_{i_h}^{i_h+1}$ **is the probability of successful transmission of a single bit from agent** i_h **to agent** i_{h+1}

Successful Transmission Probability

- The probability is given by the probability of maintaining the SNR(signal to noise ratio) at the receiver above a target level ν_0
- $\Pr_{i_h}^{i_h+1} = \exp\left(-\frac{\sigma^2 \nu_0 \left(D_{i_h, i_{h+1}}\right)^\alpha}{\kappa \tilde{P}}\right)$
  - 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}
  - $\tilde{P}$ **is the maximum transmit power of agent** i_h

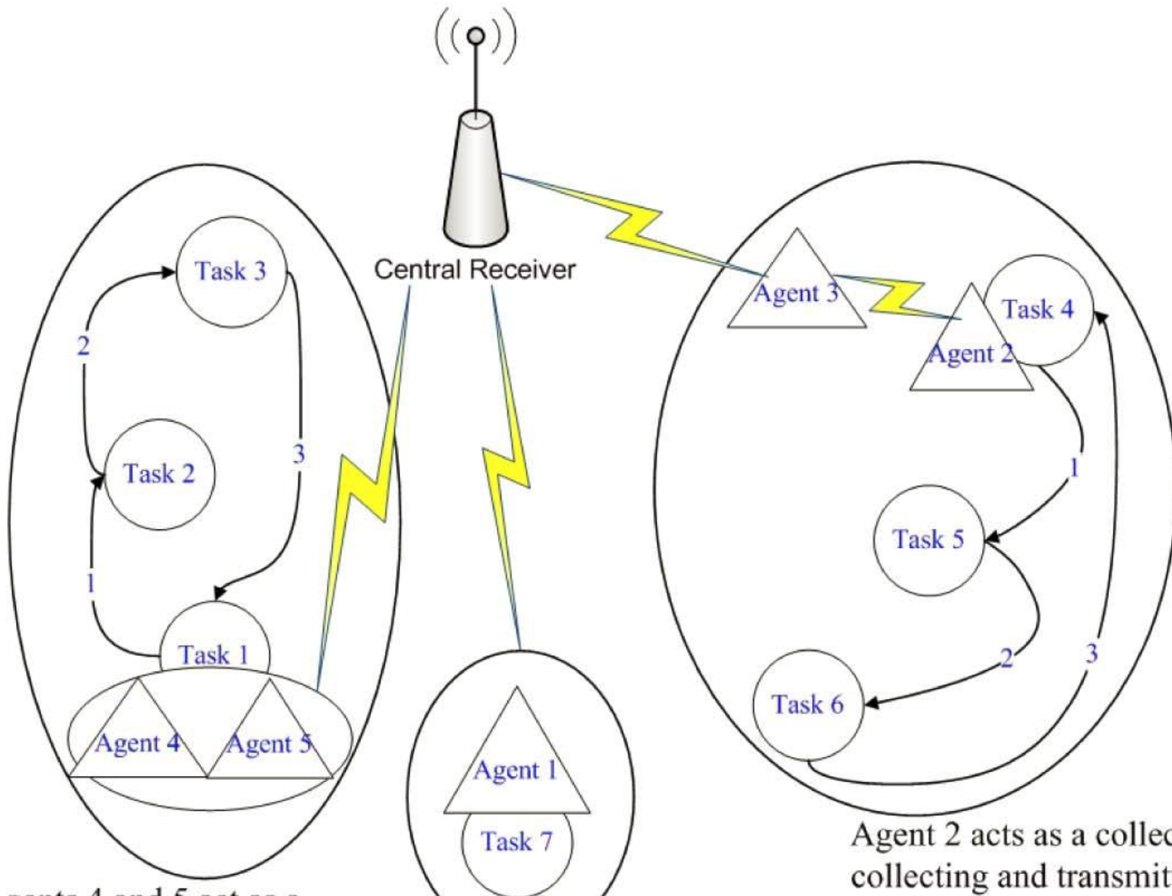
Serving Tasks

- For servicing a number of tasks $C \subseteq \mathcal{T}$, a group of agents $G \subseteq \mathcal{M}$ 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.
- 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 relays also move to connect the task being served to the central receiver

System Model

The Command Center is the intelligence that has some control over the agents

The task allocation problem among the agents can be mapped into the problem of the formation of coalitions



Agents 4 and 5 act as a single collector, collecting and transmitting data from tasks 1,2, and 3. The tasks are visited in the order 1,2,3 continuously in that order

Agent 1 acts as a collector, collecting and transmitting data from task 7

Agent 2 acts as a collector, collecting and transmitting data from tasks 4,5, and 6 through Agent 3 which acts as a relay. Both agents visit tasks 4,5,6 continuously in that order.

Coalitional Game Formation

- Model task allocation as a coalitional game with transferable utility
  - **Coalitional Game: groups of players can achieve rather than on what individual players can do.**
- Propose a suitable utility function for this model
  - **represents the total revenue achieved by a coalition.**

Game Formulation

- The task allocation coalitional game is played between the agents and the tasks.
- The players set $\mathcal{N}$ contains both agents and tasks, i.e., $\mathcal{N} = \mathcal{M} \cup \mathcal{T}$
- For any coalition $S \subseteq \mathcal{N}$
  - **agents: collectors and relays**
  - **tasks**

Polling System

- A polling system is one that contains a number of queues served in *cyclic order*.
- 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.
- The proposed task servicing scheme could be mapped as *polling system.*
- Single server
  - **The collectors of every coalition**

Polling System

- The *exhaustive strategy* for a polling system
  - **Whenever the collectors stop at any task $i \in S$, they service a queue until emptying the queue**
  - **This strategy is applied at the level of every coalition $S \subseteq \mathcal{N}$**
- *Switchover time*
  - **The time for the server to move from one queue to the other**

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 nonzero 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.

Property 2

- When move from one task to the other, assume all agents start their mobility at the same time, and move in straight line trajectories.
- $\theta_{i,j}$ denotes the swichcover time from task i to j.
- 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.

Waiting Time

- Pseudoconservation Law

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

- $\rho_S = \sum_{i \in S \cap \mathcal{T}} \rho_i$ and $\rho_i = \frac{\lambda_i}{\mu^{G_S}}$, the utilization factor of task i
  - λ_i **is the average arrival rate of each task**
  - μ^{G_S} **is the total link transmission capacity of coalition** S
  - G_S **is the collectors in coalition** S
- $\theta_s = \sum_{h=1}^{|s \cap \mathcal{T}|} \theta_{i_h, i_{h+1}}$ is the sum of switchover time

Waiting Time

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

The average queuing delay for M/D/1 queues, weighed by ρ_S

The delay resulting from the switchover period.

Conclusion: *Adding more collectors* -> increasing μ^{G_S} -> decreasing ρ_S -> *Reducing waiting time*

Stability

- For any coalition S in the system, the following condition must hold
  - $\rho_S < 1$
- This condition is a requirement for the stability of any polling system. Therefore, it's also a requirement for the stability of any coalition in the system

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 $\{i_1, i_2, \ldots, i_{|S \cap \mathcal{T}|}\}$ which is an ordering over the set of tasks in S given by $S \cap \mathcal{T}$.
- *Goal*: Minimize the total switchover time for one round of data collection
- Traveling salesman problem
  - **NP-complete**
  - **The nearest neighbor algorithm**

Utility Function

- For every coalition, the benefit, in terms of the average *effective throughput* that the coalition is able to achieve,

$$L_S = \sum_{i \in S \cap \mathcal{T}} \lambda_i \cdot Pr_{i,CR}$$

- *Adding more relays* will reduce the distances over which transmission is occurring, thus, *improving the probability of successful transmission*.
- *Throughput* or *Waiting time*?

Utility Function

- Exhibit a trade-off between the *throughput* and the *delay*
- The utility of every coalition S is evaluated using a *coalitional value function* based on the power concept

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

- $\beta \in (0,1)$ is a throughput-delay *trade-off parameter*, δ is the *price* per unit power that the network offers to coalition S.

Coalitional Game

- Consequently, given the set of players $\mathcal{N}$, and the given value function v, we define a coalitional game $(\mathcal{N}, v)$ with transferable utility (TU).
- The total achieved revenue can be arbitrarily apportioned between the coalition members.
- Equal fair allocation rule: the payoff of any player $i \in S$, denoted by x_i^S is given by

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

  - **Represents the amount of revenue that player** $i \in S$ **receives from the total revenue** $v(S)$ **that coalition** S **generates**

Task Allocation as a Hedonic Coalition Formation Game

- Hedonic Coalition Formation: Concepts & Model
- Hedonic Coalition Formation: Algorithm
- Distributed Implementation Possibilities

Hedonic Coalition Formation: Concepts & Model

- Hedonic coalition
  - **Economics; Wireless networks**
- Two key requirements for classifying a coalitional game as a hedonic game
  - **The payoff of any player depends *solely* on the numbers of the coalition to which the player belongs**
  - **The coalitions form as a result of the *preferences* of the players over their possible coalitions' set**

Coalition Partition & Player's Coalition

- Def. 1: A coalition structure or a coalition partition is defined as *the set* $\prod = \{S_1, S_2, \ldots, S_l\}$ *which partitions the players set* $\mathcal{N}$, i.e., $\forall k, S_k \subseteq \mathcal{N}$ are *disjoint coalitions* such that $\bigcup_{k=1}^{l} S_k = \mathcal{N}$.
- Def. 2: Given a partition $\prod$ of $\mathcal{N}$, for every player $i \in \mathcal{N}$, we denote by $S_{\prod}(i)$, the coalition to which player i belongs, i.e., coalition $S_{\prod}(i) = S_k \in \prod$, such that $i \in S_k$

Preference Relation & Hedonic Coalition Game

- Def. 3: For any player $i \in \mathcal{N}$, a preference relation or order $\succcurlyeq_i$ 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 $\{S_k \subseteq \mathcal{N} : i \in S_k\}$
- Def. 4: A hedonic coalition formation game is a coalitional game that satisfies the two hedonic conditions previously mentioned, and is defined by the pair $(\mathcal{N}, \succ)$ where $\mathcal{N}$ is the set of players and $\succ$ is a profile of preferences

Evaluate Preference Relation

- For agents, $S_2 \succcurlyeq_{\mathcal{M}} S_1 \Leftrightarrow u(S_2) \geq u(S_1)$, where,

$$u(S) = \begin{cases} \infty, & if S = S_{\prod}(i) \& S \backslash \{i\} \subseteq \mathcal{T} \\ 0, & if\ S \in h(i) \\ x_i^S, & otherwise \end{cases}$$

  - **Where,** $h(i)$ **is the history set of player** i

- For tasks, $S_2 \succeq_{\mathcal{T}} S_1 \Leftrightarrow w_j(S_2) \geq w_j(S_1)$

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

Bound on the number of collector-agents

- For the proposed hedonic coalition formation model for task allocation, assuming that all collector-agents have an equal link transmission capacity $\mu_i = \mu$, 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:

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

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

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

- 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.

Hedonic Coalition Formation: Algorithm

- The rule for coalition formation(Def. 5) : Given a partition $\prod = \{S_1, \ldots, S_l\}$ of the set of players (agents and tasks) $\mathcal{N}$, a player i decides to *leave* its current coalition $S_{\prod}(i) = S_m$, for some $m \in \{1, \ldots, l\}$ and *join* another coalition $S_k \in \Pi \cup \{\emptyset\}$, if and only if $S_k \cup \{i\} \succ_i S_{\Pi}(i)$. Hence, $\{S_m, S_k\} \rightarrow \{S_m \backslash \{i\}, S_k \cup \{i\}\}$.
- Starting from any initial network partition 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.*

Nash-stable & Partition Stable

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* $\succeq_i = \succeq_{\mathcal{T}}, \forall i \in \mathcal{N} \cap \mathcal{T}$*).*

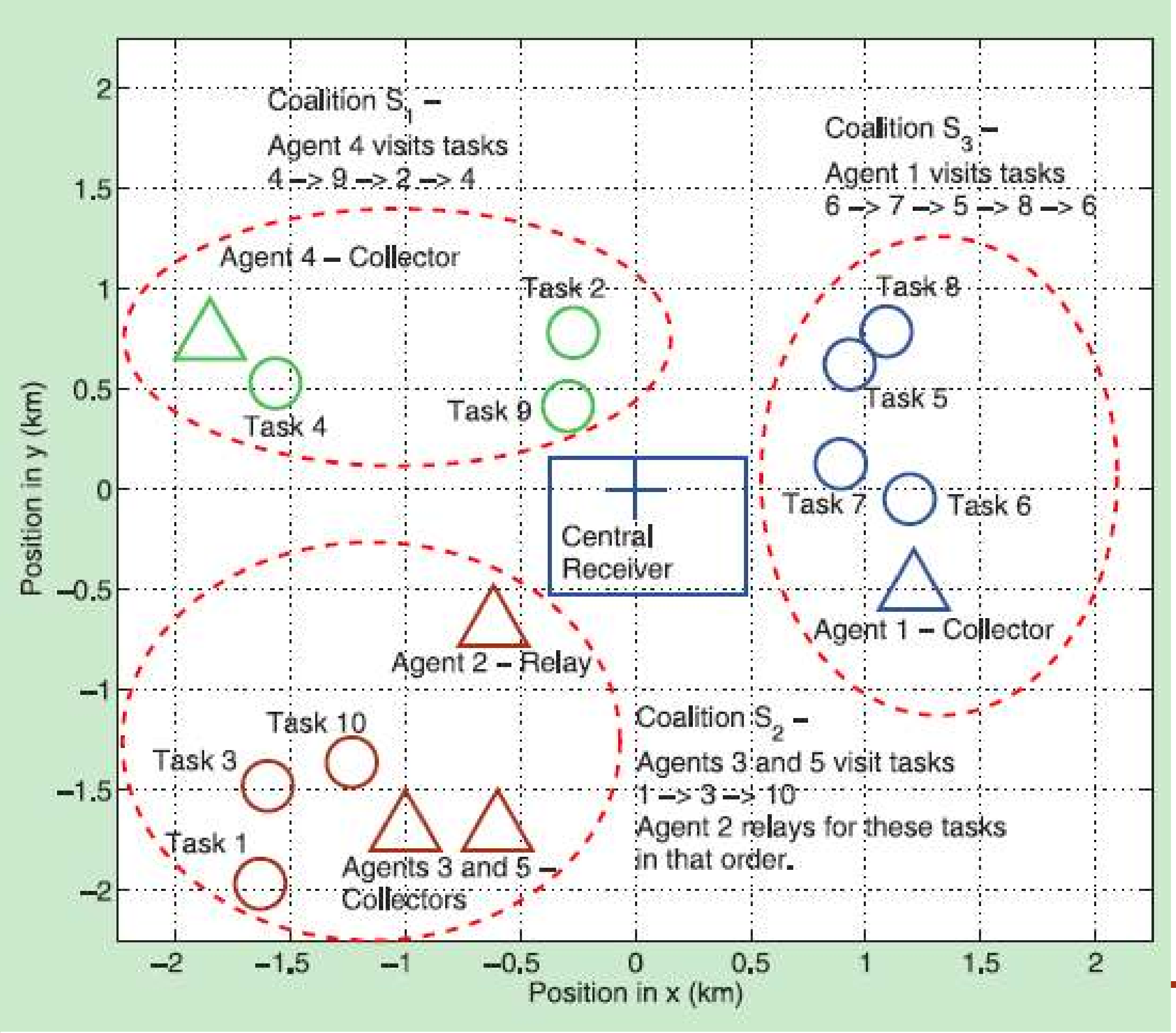
Definition 7. *A partition* $\Pi = \{S_1, \ldots, S_l\}$ *is* individually stable *if there do not exist* $i \in \mathcal{N}$, *and a coalition* $S_k \in \Pi \cup \{\emptyset\}$ *such that* $S_k \cup \{i\} \succ_i S_{\Pi}(i)$ *and* $S_k \cup \{i\} \succeq_j S_k$ *for all* $j \in S_k$ *(for agents* $\succeq_i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

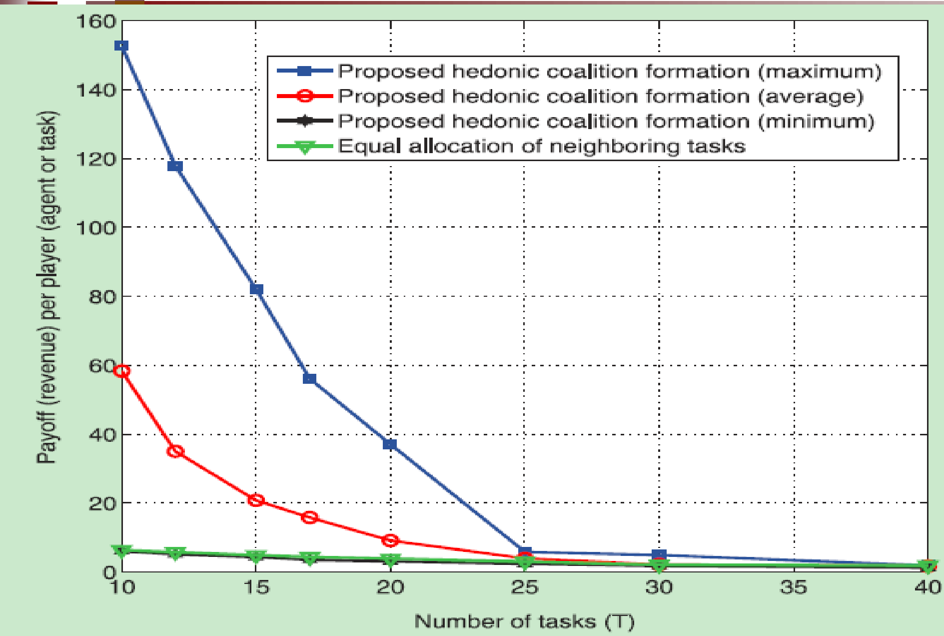
- Any partition Π_f resulting from the hedonic coalition formation phase of the proposed algorithm is Nash-stable, and hence individually stable.

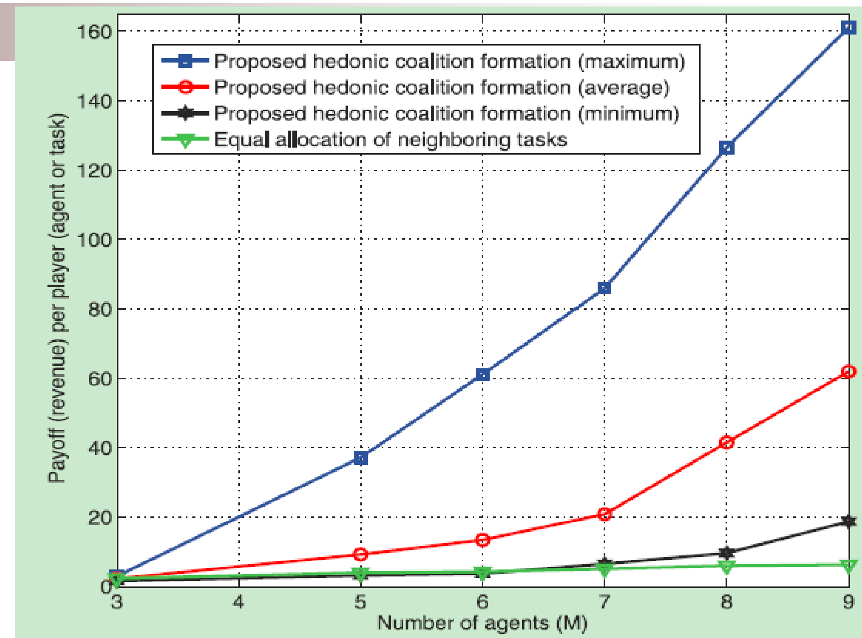
Distributed Implementation Possibilities

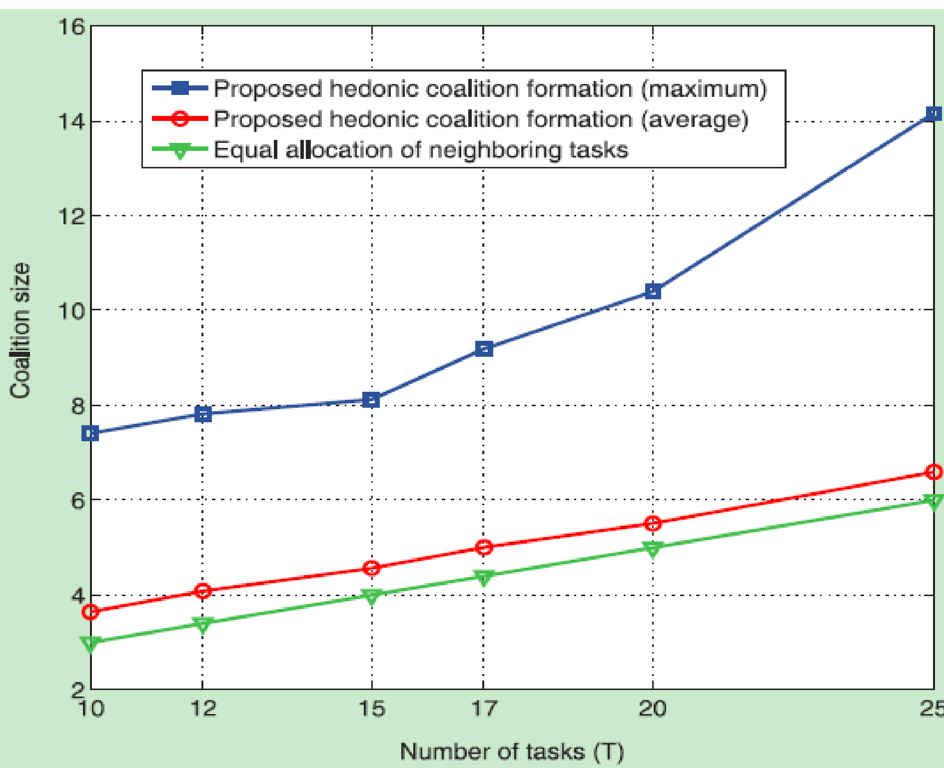
- Command server and Central receiver
- Information for performing coalition formation
  - **Agents: Location and arrival rate of tasks**
    - Tasks' owner -> Command Center -> Databases
    - Agents could access databases to get the information
  - **Tasks: the actual presence of agents**
    - Agents announce/broadcast their presence to the tasks
- 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 <u>independently of any centralized entity</u>

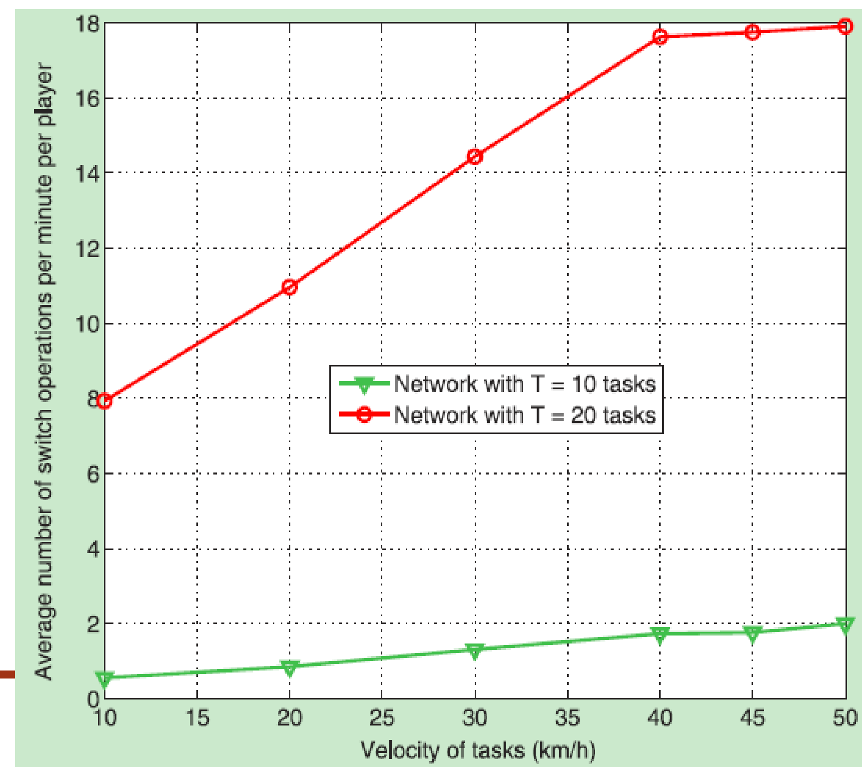
Simulation Results & Analysis

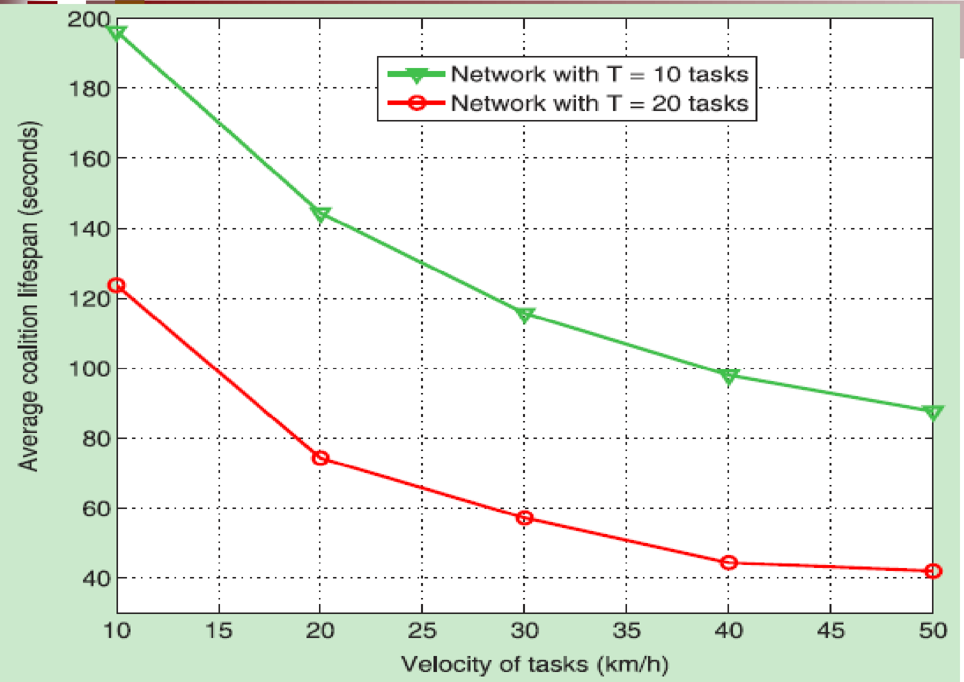


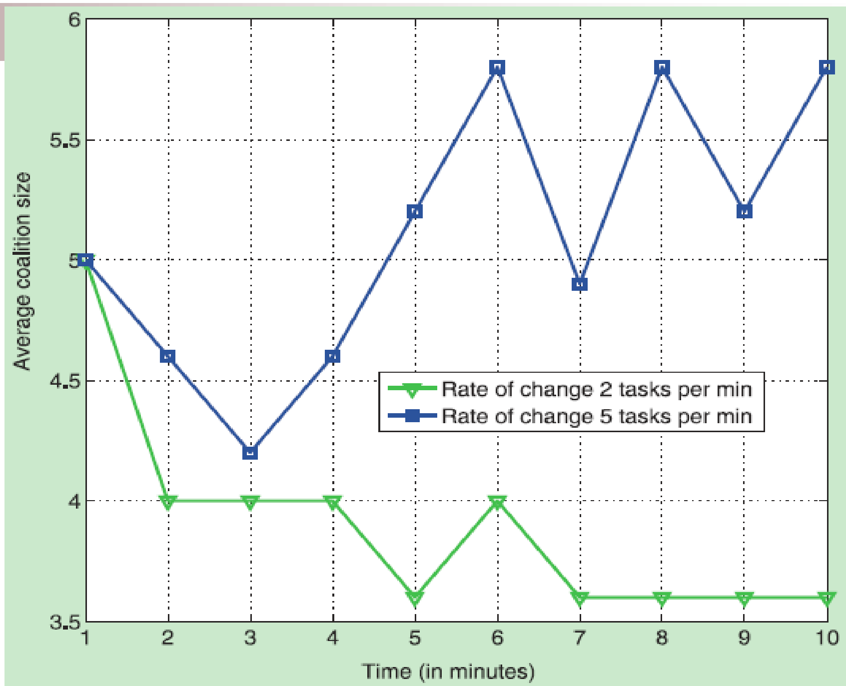

Proposed hedonic coalition formation (maximum)
Proposed hedonic coalition formation (average)
Proposed hedonic coalition formation (minimum)
Equal allocation of neighboring tasks
Payoff (revenue) per player (agent or task)
Number of tasks (T)
0
20
40
60
80
100
120
140
160
10
15
20
25
30
35
40

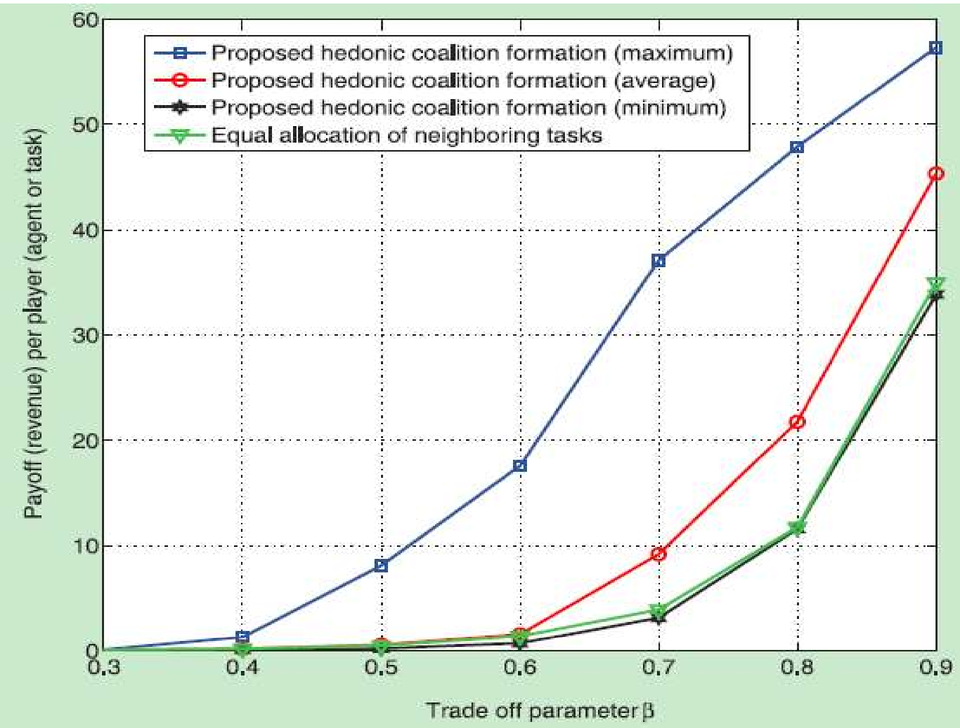

Proposed hedonic coalition formation (maximum)
Proposed hedonic coalition formation (average)
Proposed hedonic coalition formation (minimum)
Equal allocation of neighboring tasks
Payoff (revenue) per player (agent or task)
Number of agents (M)
0
20
40
60
80
100
120
140
160
3
4
5
6
7
8
9


Proposed hedonic coalition formation (maximum)
Proposed hedonic coalition formation (average)
Equal allocation of neighboring tasks
Coalition size
Number of tasks (T)
2
4
6
8
10
12
14
16
10
12
15
17
20
25

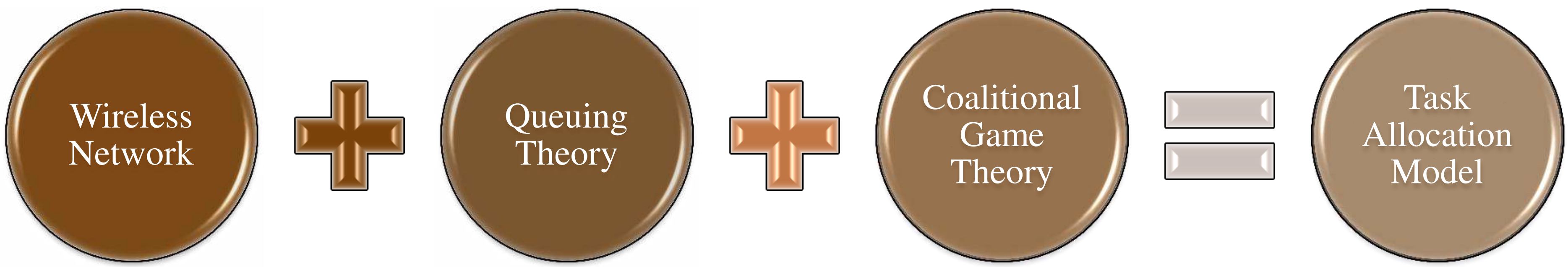

Network with T = 10 tasks
Network with T = 20 tasks
Average number of switch operations per minute per player
Velocity of tasks (km/h)
0
2
4
6
8
10
12
14
16
18
10
15
20
25
30
35
40
45
50


Network with T = 10 tasks
Network with T = 20 tasks
Average coalition lifespan (seconds)
Velocity of tasks (km/h)


Rate of change 2 tasks per min
Rate of change 5 tasks per min
Average coalition size
Time (in minutes)


Proposed hedonic coalition formation (maximum)
Proposed hedonic coalition formation (average)
Proposed hedonic coalition formation (minimum)
Equal allocation of neighboring tasks
Payoff (revenue) per player (agent or task)
Trade off parameter β

Conclusions




Thank You

References

- M. Debbah, "Mobile Flexible Networks: The Challenges Ahead," Proc. Int'l Conf. Advanced Technologies for Comm., Oct. 2008.
- J. Proakis, Digital Communications, fourth ed., McGraw-Hill, 2001.
- H. Takagi, Analysis of Polling Systems. MIT, Apr. 1986.
- H. Levy and M. Sidi, "Polling Systems: Applications, Modeling, and Optimization," IEEE Trans. Comm., vol. 38, no. 10, pp. 1750-1760, Oct. 1990.
- Y. Li, H.S. Panwar, and J. Shao, "Performance Analysis of a Dual Round Robin Matching Switch with Exhaustive Service," Proc. IEEE Global Telecomm. Conf., Nov. 2002.
- V. Vishnevsky and O. Semenova, "The Power-Series Algorithm for Two-Queue Polling System with Impatient Customers," Proc. Int'l Conf. Telecom., June 2008.

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.