Game Theory Final Project

Gateway Selection Game in Cyber-Physical Systems

Alessandro Toschi - 118033990016
Edoardo Cappello - 118036990002
Giulia Serafini - 118036990003

1. Introduction

Nowadays IOT technologies are primarily adopted in many contexts, and Cyber-Physical Systems (CPSs) obtain data from the physical world and influence the environment with different kind of devices. The most important feature needed by this kind of systems is the ability of their components to communicate among them and with external networks, usually dealing with different and incompatible wireless technologies. Thus, the direct connection of the devices with external networks is not possible, and the use of gateways has become necessary.
This necessity of communication among different technologies and the availability of connection for all the devices of the CPS has led to intensively placing this kind of devices. This fact implies that often the coverage areas of different gateways overlap, so it is normal for a device to choose among different gateways to find the best one for them to keep a connection.
Usually, devices in CPSs consider just their benefit, without cooperating. The competition becomes even more intense if we consider different kinds of devices. However, the cooperation among the same kind of devices is an assumption that can be done, because the same kind of devices may transmit data for the others and usually they have a similar responsibility.
In [1], the gateway selection problem is modeled as a noncooperative game competing for the gateway's bandwidth, and a practical bandwidth arrangement method is proposed. Furthermore, the migration trend is studied when every device chooses a new gateway to maximize the total benefit of its group's type. The convergence of this game is not guaranteed, and some theorems are proposed to describe better the condition for the existence of a Nash Equilibrium.

2. Our Work

Our project consists in:

- analyzing deeply the paper proposed by Hao Wang et al. [1];
- finding the game theory model related to the proposed problem;
- implementing a simulator in python to examine the dynamics of the gateway selection game.

For having a live experience with it, it is possible to find the source code at the following link: ***https://github.com/giuliaserafini/GameTheoryProject***

3. Model Adopted

The *Network Model* is made up of three different kind of nodes and no central controller. The first group is represented by the Gateway nodes, $G = \{g_1, g_2, \ldots, g_k\}$; meanwhile, the other two groups are the Client nodes $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_m\}$, each composed by a list of devices. Each device of each client has to be connected to one and only one gateway, and the same kind of devices (i.e. they belongs to the same client) can exchange data mutually. Moreover, a device can obtain nearby gateways information which contains number and type of devices connected to this gateway. With this information, every device can calculate the total bandwidth it will get by changing the gateway to which be connected.

The *Bandwidth Allocation Model* consists in dividing the resources equally: every gateway g_i has a total bandwidth W, which is arranged to clients A and B by W_{a_i} and W_{b_i}, with $W_{a_i} + W_{b_i} = W$. In each gateway, the devices of the same type share the assigned bandwidth equally:
$w_a = \frac{W_{a_i}}{n_i}$ and $w_b = \frac{W_{b_i}}{m_i}$,
where w_a and w_b are respectively the bandwidth obtained by a single device type a and b, n_i represents the number of devices type A connected to the gateway g_i, and m_i represents the number of devices type B connected to the gateway g_i.
The total amount of bandwidth allocated for a client type *T* is proportional to the number of the devices type *T* connected, so $W_{a_i} : W_{b_i} = n_i : m_i$. Therefore, the total bandwidth assigned to a client a and a client b by a gateway g_i is:

$$w_{a,i} = \frac{n_i \left(\frac{W}{n_i + m_i}\right)}{n_i} = \frac{W}{n_i + m_i}$$

$$w_{b,i} = \frac{m_i \left(\frac{W}{n_i + m_i}\right)}{m_i} = \frac{W}{n_i + m_i}$$

From the two previous formulas, we can notice that the bandwidth each client depends just on W, n_i and m_i.

The *Gateway Selection Game* is based on clients which want to maximize their benefit.
Each device can exchange data with the same type of devices. If a device increases its bandwidth, then all the devices of the same kind will have a benefit in it. Therefore, a device will change gateway if and only if this migration will increase the total bandwidth of its group.
The game can be formalized as follows:

- *Player*: a client who can connect to more than one gateway in the working area (each client i is composed by a list of devices, i.e. the client's population).
- *Strategy*: each player (client) i has a strategy $S_i = \{s_{i_1}, s_{i_2}, \ldots, s_{i_n}\}$, and each device d type i has its strategy set $s_{i,d} \subseteq G$ which includes all the possible gateways reachable by d.
- *Payoff*: the bandwidth obtained by a client.
- *Population*: devices from the same kind of clients. In our case, we will refer to them as A and B.
- *Nash Equilibrium*: a strategy profile is a Nash Equilibrium if none of the players can increase its population's payoff by changing to another strategy, while the other players are not modifying their ones.

4. Mathematical Formulation

Sets

$A = \{a_1, a_2, \ldots, a_n\}$ *is the set of the devices type A*
$B = \{b_1, b_2, \ldots, b_m\}$ *is the set of the devices type B*
$P = \{A, B\}$ is the set of the players, where A and B are respectively *client A* and *client B*
$G = \{g_1, g_2, \ldots g_k\}$ is the set of the *k-available gateways*

Strategies

$S_A = \{s_{a_1}, s_{a_2}, \ldots, s_{a_n}\}$, where s_i is the strategy of the device $a_i \in A$, i.e. $s_i \subseteq G$ and contains all the possible gateways reachable by a_i

$S_B = \{s_{b_1}, s_{b_2}, \ldots, s_{b_m}\}$*, where* s_i *is the strategy of the device* $b_i \in B$*, i.e.* $s_i \subseteq G$ *and contains all the possible gateways reachable by* b_i

Decision Variables

$$x_{i,g} = \begin{cases} 1 & \text{if the device i selects the gateway g and g} \in s_i \\ 0 & \text{otherwise} \end{cases} \qquad i \in A \cup B, g \in G$$

n_g = number of devices type A connected to gateway g, i.e. $n_g = \sum_{i=1}^{n} x_{a_i,g} \quad a_i \in A, g \in G$

m_g = number of devices type B connected to gateway g, i.e. $m_g = \sum_{i=1}^{m} x_{b_i,g} \quad b_i \in B, g \in G$

$w_{i,j}$ = bandwidth obtained by player i connected to gateway g,

i.e. $w_{i,g} = \begin{cases} \frac{W}{n_g + m_g} & \text{if } n_g + m_g > 0 \\ 0 & \text{otherwise} \end{cases} \qquad i \in P, g \in G$

W_A = total bandwidth obtained by player A, i.e. $W_A = \sum_{g=1}^{k} w_{A,g}$

W_B = total bandwidth obtained by player B, i.e. $W_B = \sum_{g=1}^{k} w_{B,g}$

Payoffs

W_A and W_B are respectively the payoffs of player A and player B.

Assumptions

- each device must be connected to one and only one gateway
- each device does not have a minimum bandwidth's demand
- all the gateways have the same bandwidth *W*

Nash Equilibrium

The game reaches the Nash Equilibrium when each player selects the strategy which maximizes its payoff:

$$\max \quad W_A + W_B$$

s.t.

$$\sum_{g=1}^{k} x_{i,g} = 1 \quad \forall i \in A \cup B$$ (each device must be connected to one and only one gateway)

$$W_A = \sum_{g=1}^{k} w_{A,g}$$ (the total bandwidth of player A is the sum of the bandwidth given by each gateway)

$$W_B = \sum_{g=1}^{k} w_{B,g}$$ (the total bandwidth of player B is the sum of the bandwidth given by each gateway)

where $w_{i,j}$ has been previously defined.

5. GATEWAY SELECTION ALGORITHM

The dynamic of the gateway selection game, i.e. the way in which every device decide if migrating to another gateway, can be seen as an algorithm.
The following pseudo-code describes the way in which one device performs its gateway choice:

Input	• Device's accessible gateways set G_i • Device's current gateway g • Parameter of each gateway g_i in G_i (bandwidth) • Number of devices of type A and devices type B connected to g_i
Output	• Decision to migration • Eventually, the new gateway selected for the migration
Algorithm	1. **for each** gateway g_i **in** G_i 2. calculate the increment Δw_i of the total payoff of the population after migration 3. **if** $\Delta w_i > 0$ 4. device changes its gateway to g_i 5. **else** 6. device keeps its current gateway

The algorithm is quite simple. The only thing not yet explained is how to calculate the increment Δw_i of the total payoff of the population after migration, that is necessary for choosing if migrating to the new gateway. In the following formula, we assume that g_0 is the actual gateway and g_i is the gateway evaluated for the migration. All the data are subscripted with the respective gateway number. Note that all the used data used are available as inputs of the algorithm.
If the device type T wants to change its gateway from g_0 to g_i, the total payoff will be:

$$\Delta w_i = \left[(n_0 - 1)\frac{W}{n_0 + m_0 - 1} \; + \; (n_i + 1)\frac{W}{n_i + m_i + 1} \right] \; - \; \left(n_0 \frac{W}{n_0 + m_0} \; + \; n_i \frac{W}{n_i + m_i} \right)$$

where n_j is the number of devices type T and m_j is the number of devices of the other type, referring to the gateway g_j.

There are some particular cases which do not allow to solve the previous equation. For each of them a slightly modified formula is proposed:

- $n_0 + m_0 = 1$ and $n_i + m_i = 0$, which means $n_0 = 1$ and $n_i = m_i = 0$:

  $$\Delta w_i = \left[(n_i + 1)\frac{W}{n_i + m_i + 1} \right] - \left(n_i \frac{W}{n_i + m_i} \right) = 0$$

  so the client will not change the gateway.

- $n_0 + m_0 \geq 2$ and $n_i + m_i = 0$:

  $$\Delta w_i = \left[(n_0 - 1)\frac{W}{n_0 + m_0 - 1} \; + \; (n_i + 1)\frac{W}{n_i + m_i + 1} \right] \; - \; \left(n_0 \frac{W}{n_0 + m_0} \right) > 0$$

so the client will move to the gateway g_i.

- $n_0 + m_0 = 1$ and $n_i + m_i \geq 1$:

$$\Delta w_i = \left[(n_i + 1)\frac{W}{n_i + m_i + 1}\right] - \left(n_0\frac{W}{n_0 + m_0} + n_i\frac{W}{n_i + m_i}\right) < 0$$

so the client will not change the gateway.

For a better understanding the algorithm's dynamics, the authors proposed the following theorems [1].

Theorem 1. *If the number of clients A and clients B of the original gateway equals the number of clients A and clients B of candidate gateway, respectively, none of these clients will change its strategy.*

Theorem 2. *If the number of one kind of clients in the original gateway is equal to that in the candidate gateway, the other kind of clients in original gateway will change its strategy iff there are at least two clients more in original gateway than in candidate gateway.*

Theorem 3. *Denote the number of clients B in g_1 and g_2 by n_1 and n_2, and denote the number of clients B in g_1 and g_2 by m_1 and m_2. When $n_1 = n_2 = n$ and $m_1 \neq m_2$, we have the following.*

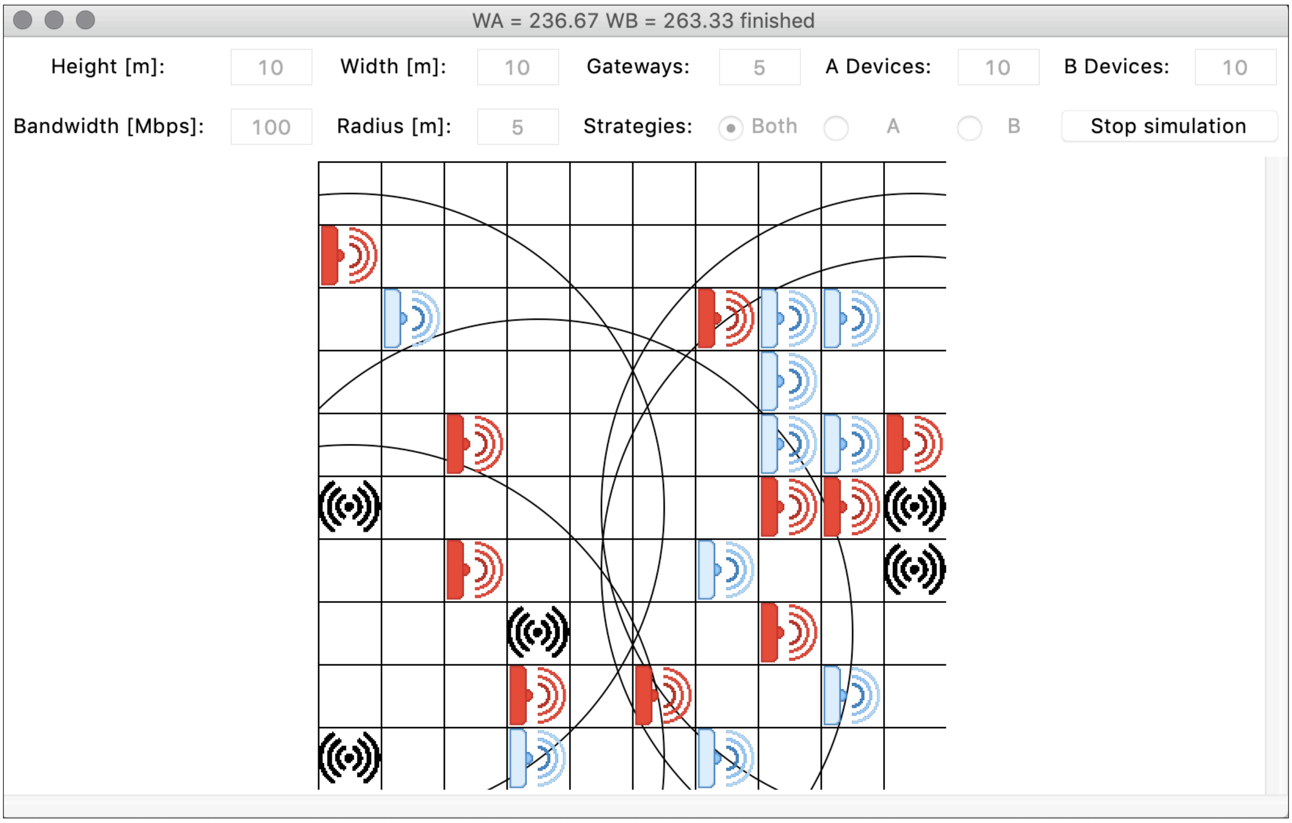
1. *If $m_1 = m_2 + 1$, none of players of clients A in g_1 will change its strategy to g_2.*
2. *If $m_1 < m_2$ and $n^2 < m_1 \cdot m_2$ or $m_1 < m_2$ and $n^2 > m_1 \cdot m_2$, none of players of clients A in g_1 will change its strategy to g_2.*

An important problem which can happen with this algorithm is the convergence: in some cases, a loop can occur, i.e. a subset of devices continue performing the same sequence of moves because of each others' moves. Also for this kind of problem, a theorem has been proposed:

Theorem 4. *If only one kind of clients can change their strategies, this gateway selection game will always come to a Nash Equilibrium.*

However, theorem 4 refers to a more constrained problem. In our implementation of the algorithm, we found a way for establishing if we are in a loop situation. More details will be provided in the next section.

6. SIMULATION



We developed a simulator, written in python, which implements the *Gateway Selection Algorithm.* Our model is composed of devices, gateways and space which represents a two-dimensional area in which is possible to place devices and gateways.

A device is a primary element that has to respect two constraints:

- It can connect to *one and only one* gateway;
- It must be placed in a position which allows itself to reach at least one gateway.

The gateways enforce the first constraint: they do not allow a device to connect if it is already connected to someone else.

The elements are placed in the space in order, first the gateways and then the devices.
The order is relevant because we did not assume the gateways are uniformly distributed in the space. To simulate a real scenario, we randomly put the gateways. Then, to respect the second constraint, each device can be positioned if it reaches at least one gateway. Indeed, the space is not just a passive container of elements, but it plays a key role in the geometry definition of the problem and makes it necessary.

The algorithm is lead by the devices: they choose to connect to a gateway according to their strategies. The algorithm proposed in the paper is distributed, but in our simulation, we preferred a sequential approach that does not affect the generality of the problem and the execution's results.

We spent some effort trying to find out a solution for the non-termination of the *Nash Equilibrium* when both devices can change their strategies.
We noticed that sometimes there is a point in which the continuous execution of the algorithm does not affect anymore the players' strategies, in term of payoffs' maximization.
We identify as cycle the situation in which, for two consecutive rounds, the same number of devices change their strategies without incrementing the players' payoffs.

A practical solution is to count the number of consecutive cycles and force the termination of the algorithm when a threshold is exceeded.
The proposed solution simulates environments with up to hundreds elements in very few seconds, (the threshold has been set to 10 consecutive cycles) and always converges to an equilibrium.

We provided a GUI for our simulation, in which the user can choose the parameters of the simulation and follow the evolution of the network. The default parameters provide a useful and fast explanation of the problem, but the user can choose different ones, taking into account that the evolution time grows with the numbers of devices.

7. Conclusions

The gateway selection problem treated in our analysis uses basic concepts from the game theory field to solve a problem that nowadays is more and more common, i.e. the distribution of devices in a network of gateways.
Since the game can be considered non-cooperative, it has been possible to implement an algorithm which allows a device to establish which is the best gateway to be connected.
By setting the possible strategies and payoffs of a class of devices, it is possible to adapt the problem to a typical game theory scenario, exploiting the already well-known techniques that this kind of framework offers.
Moreover, we implemented an algorithm in python which simulates the behavior of two populations of devices that are competing for obtaining the highest payoff (i.e. gateways' bandwidth). We also discussed the problem of the termination of the game, and we found how to detect loop situations and how to enforce a pseudo-equilibrium.

8. References

[1] Hao Wang, Jianzhong Li, and Hong Gao, “Gateway Selection Game in Cyber-Physical System”, International Journal of Distributed Sensor Network, Vol. 2016, Atricle ID 7190767