Matching Theory Application for Efficient Allocation of Indivisible Testbed Resources

Donatos Stavropoulos[†⋆], Vasileios Miliotis[†], Thanasis Korakis[†⋆]
and Leandros Tassiulas[‡]
[†]Department of Electrical and Computer Engineering, University of Thessaly, Greece
[⋆]Centre for Research and Technology Hellas, CERTH, Greece
[‡]Department of Electrical Engineering, Yale University, New Haven, USA
Email: {dostavro, vmiliotis, korakis}@uth.gr, leandros.tassiulas@yale.edu

Abstract—**In this paper, we examine the problem of the efficient allocation of resources in networking testbeds, which cannot be shared among the experimenters. We highlight the similarities with the housing market where indivisible network resources play the role of houses, while experimenters the role of owners. We adopt the Top-Trading-Cycles (TTC) algorithm for providing Pareto efficient allocations and we compare this approach with the current mechanism of the simple First-Come-First-Served (FCFS) approach used in most networking testbeds. A formulation of the problem is provided where we describe the average utility of the system as a function of the desired testbed resources of the experimenters and the final allocation of the resources to them. In the performance evaluation we observe that TTC outperforms FCFS in all the examined scenarios and achieves almost 95% better average utility in certain cases.**

I. INTRODUCTION

Networking experimentation testbeds have been developed around the world in order to satisfy the needs for real life large scale experimentation by initiatives and projects like GENI [1] in US and FIRE [2] in Europe. Lately these efforts have been transformed to the provision of infrastructure capable of 5G experimentation originated by the funding of 5G-PPP programme [3], which fosters the collaboration of academia and industry on the research and advancement of 5G technology. While the support and funding for experimentation facilities present high dependency on public funding [4], mainly from H2020 research projects in Europe and the PAWR program [5] in US, the increasing demand for experimentation on buzzword technologies like 5G and IoT creates pressure for testbed sustainability. As the capital expenditure for testbed advancements follows the pace of research project funding, the efficient allocation of testbed resources is of prominent importance. Academic and industry research groups increasingly seek access to cutting-edge equipment in order to evaluate their concepts and frameworks in realistic environments and obtain valuable measurements. This fact leads to an increasing number of users in networking testbeds and a high competition among the community for the reservation of available testbed resources.

Testbed owners have been in a constant effort of upgrading their management frameworks by following the latest trends on testbed federations, which allow them to be part of large-scale federated testbeds that experimenters from around the world can use with the same set of tools, like in Fed4FIRE [6]. However, in these efforts there is no focus on optimizing the resource allocation of the provisioned resources, since most of the testbeds work in a best effort manner, meaning that the testbeds' reservation management frameworks provide access to resources as long as there is availability in a First-Come-First-Served (FCFS) priority.

The same principles apply to our own networking testbed NITOS Future Internet experimental facility [7], which provides resources that cover most of the wireless and wired network trending technologies. In more detail, NITOS operates 24/7 all year and offers remote access for wireless experimentation to those who want to deploy and experimentally evaluate networking protocols and applications in real world settings, incorporating technologies that include and are not limited to WiFi, 4G, 5G, IoT, SDN, SDR, Cloud Computing and more. For the management of the daily operations of the testbed we have built a management framework called NITOS Broker [8], which is responsible for the discovery, reservation and provisioning of the testbed resources to experimenters requesting access. In this context, the Broker can provide a list of the available resources to the experimenters in a calendar format, allowing them to reserve in a FCFS manner the resources they choose.

Taking into consideration the problems of overuse of resources and unfairness of the current reservation mechanism, we came up with a more efficient and fair scheme that increases the total utility of the system by not compromising ease of use for the experimenters. In this paper, we present our solution which will replace the existing mechanism implemented in the NITOS Broker. Our proposed mechanism stems from the domain of housing market where the exchange of indivisible resources among owners is conducted without the use of money. The similarities with the domain of experimentation testbeds where experimenters try to obtain indivisible resources without any form of currency, allow us to apply a slightly modified solution to an algorithm used in the housing market.

The rest of the paper is organized as follows: In Section II related work is discussed and in Section III the problem is analyzed, resulting in a system model formulation. In Section IV an overview of the algorithms for resource matching is presented and in Section V the evaluation of the proposed

978-1-7281-4973-8/20/$31.00 © 2020 IEEE

scheme is provided. Finally, in Section VI we conclude our work and provide our future plans.

II. RELATED WORK

In the domain of resource allocation in networks many studies have been published, but very few were focused on allocating resources in networking testbeds. To our knowledge, none had approached the problem from the socioeconomic aspect of matching resources to experimenters as in the market of houses and owners.

An initial approach on improving the reservation mechanism of NITOS has been proposed in [9], looking at the problem from an auction based perspective. The proposed solution enabled the experimenters to provide their actual valuations regarding the requested resources and introduced a closed virtual economy/point system that allowed the experimenters to bid on the resources they wanted. A scheme in the domain of auctions has been studied in [10] as well, in which a fist-price auction was utilized for solving the problem of resource reservation among the experimenters of a wireless testbed. Our approach differentiates from auction schemes like the aforementioned and the one proposed in [11], since it does not require a virtual currency and does not introduce extra implementation overhead that a charging/auction system requires. The experimenters do not have to think how much they will bid on the resources or worry about running out of any virtual currency.

To our best knowledge, most of the efforts on resource allocation are focused on virtualised networking resources [11], [12], [13] and none of them is focused on indivisible resources offered by testbeds to experimenters. The aforementioned studies focus on allocating divisible resources like bandwidth, CPU power, VMs etc. by formulating and solving complex optimization problems that often require heuristic solutions of NP-hard time complexity problems [14]. To the best of our knowledge, due to our involvement in several EU and US testbed development initiatives, these virtualization solutions have not yet been deployed to the testbed management systems. In our work, we focus on indivisible resources like the wireless nodes which can be employed as WiFi APs, WiFi stations or LTE Base Stations and User Equipment (UE) devices. Additionally, our approach which originates from matching algorithms for the housing market domain, requires polynomial time complexity to reach a stable allocation.

III. SYSTEM MODEL

A. Experimenting with Preferences of Equivalent Nodes

We consider an experimentation testbed with a set of nodes $\mathcal{H} = \{h_1, h_2, ..., h_N\}$ that can be reserved by experimenters in a time-slot based manner, with N being the number of the testbed's nodes. We focus on a specific experimentation time slot, for which a set of experimenters $\mathcal{A} = \{a_1, a_2, ..., a_M\}$ have declared their preferences a priori up to a deadline before the start of the slot. Each experimenter chooses a set of nodes $\mathcal{P}_a \subseteq \mathcal{H}$ that will be needed for an experiment in a preference list. After the deadline of nodes preference declaration, the testbed broker, which is responsible for allocating testbed resources to the experimenters, runs a random endowment of one node per experimenter, leading to an initial endowment e, through which one node of the set $\mathbf{P} = \mathcal{P}_{a_1} \cup \mathcal{P}_{a_2} \cup ... \cup \mathcal{P}_{a_M}$ is allocated to each experimenter $a \in \mathcal{A}$. Based on the market setup $\langle \mathcal{H}, \mathcal{A}, e \rangle$, the broker runs the Top Trading Cycles algorithm (TTC) [15], to allocate the first nodes to each experimenter. The algorithm runs $\max(|\mathcal{P}_{a_i}|)$ times, with $i = 1, 2, ..., M$ and $|\mathcal{P}_{a_i}|$ being the cardinality of $\mathcal{P}_{a_i}$, leading to a node allocation of the set $\overline{\mathcal{P}}_{a_i}$ for each experimenter a_i, $i = 1, 2, ..., M$.

We consider that the experimenters' satisfaction follows the law of diminishing marginal utility, which has been adopted in the paper as the dominant economic approach in problems that tackle allocation of resources to human beings [16], [17]. As the number of requested nodes per experimenter may vary, we express the utility function of its experimenter in a normalized form as follows:

$$U(a_i) = \frac{ln\left(1 + \frac{J_{a_i}}{1+K_{a_i}}|\overline{\mathcal{P}}_{a_i}|\right)}{ln(1 + |\mathcal{P}_{a_i}|)} \tag{1}$$

where $|\overline{\mathcal{P}}_{a_i}|$ represents the cardinality of $\overline{\mathcal{P}}_{a_i}$ and J_{a_i} is the Jaccard similarity coefficient [18] that measures the similarity between the set of initial experimenter's choices $\mathcal{P}_{a_i}$ and the set of its allocated nodes $\overline{\mathcal{P}}_{a_i}$, resulting from the broker's allocation based on the TTC. The Jaccard similarity coefficient is expressed as:

$$J_{a_i} = \frac{|\mathcal{P}_{a_i} \cap \overline{\mathcal{P}}_{a_i}|}{|\mathcal{P}_{a_i} \cup \overline{\mathcal{P}}_{a_i}|} \tag{2}$$

where $J_{a_i} \in [0, 1]$ for each experimenter a_i. With K_{a_i} we represent the total number of affected node choices of other experimenters, resulting from the allocation $\overline{\mathcal{P}}_{a_i}$, and it is expressed as:

$$K_{a_i} = \sum_{j \neq i} |\overline{\mathcal{P}}_{a_i} \cap \mathcal{P}_{a_j}| \tag{3}$$

with $j = 1, ..., M$ and $j \neq i$. Consequently, an experimenter's utility is positively affected by the level of satisfaction of its initial requests. At the same time, an experimenter's utility is negatively affected as K_{a_i} increases, reflecting the level of externality of an experimenter's satisfied choices to the interests of other experimenters. With our approach, we aim to allocate the testbed resources providing a solution to the problem described as follows:

$$\begin{aligned} \max_{\overline{\mathbf{P}}} \quad & \frac{1}{M} \sum_{i=1}^{M} U(a_i) \\ \text{subject to} \quad & |\overline{\mathbf{P}}| \leq |\mathcal{H}| \\ \text{and} \quad & |\mathcal{P}_{a_i}| > 0, \forall i = 1, ..., M \end{aligned} \tag{4}$$

B. Experimenting with Weighted Preferences

The system model presented in Section III-A refers to experiments where nodes are considered equivalent, as for example in ad hoc network experiments. We extend our study into more complex scenarios, where hierarchy of nodes is needed, such as in experiments including an LTE base station and a number of UEs. To this direction we enhance our

exp1	exp2	exp3	exp4
node 3	node 4	node 1	node 3
node 2	node 1	node 4	node 2
node 4	node 2	node 3	node 1
node 1	node 3	node 2	node 4

TABLE I: Experimenters' Preferences

system model to include different sets of preferred nodes per experimenter. In this case each experimenter $a \in \mathcal{A}$ has declared its preference sets $P_{a,l}$, for $l = 1, ..., L$, with L being the number of hierarchical groups of nodes, requested for its experiment. The requests of experimenter a_i are represented as $\mathbf{P_{a_i}} = \mathcal{P}_{a_i,1} \cup \mathcal{P}_{a_i,2} \cup ... \cup \mathcal{P}_{a_i,L}$. The total experimenters' requests for the time slot of interest are described by the superset $\mathbf{P} = \mathbf{P_{a_1}} \cup \mathbf{P_{a_2}} \cup ... \cup \mathbf{P_{a_M}}$.

To reflect the different levels of necessity of nodes of each group, we assign weights $w_{i,l}$ that represent the importance of nodes belonging to group l to the experimenter a_i, with $\sum_{l=1}^{L_i} w_{i,l} = 1$. The utility for each experimenter, after the broker has run and assigned nodes, is expressed as:

$$U(a_i) = \sum_{l=1}^{L} w_{i,l} U(a_i, l) \qquad (5)$$

where $U(a_i, l)$ is the utility described in equation (1), and represents the utility of experimenter a_i for nodes belonging to group l. We aim to allocate the testbed resources for cases of experimentation with weighted preferences, providing again a solution to the problem described in equation (4).

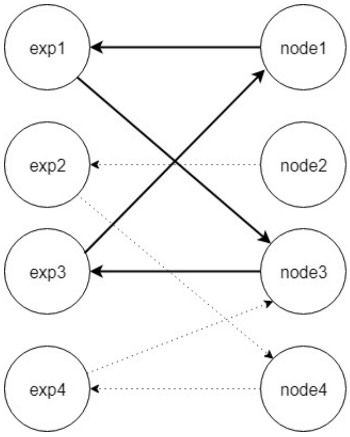


Fig. 1: TTC 1st Iteration

IV. RESOURCE MATCHING PROBLEM

The problem of allocating nodes to experimenters has significant similarities with the problem of the housing market, where individuals already own a single house and have preferences over the rest of the houses. This problem of trading indivisible items without using money, has been investigated in [15], where the Top Trading Cycles algorithm (TTC) has been proven to provide Pareto efficiency and lead to a core-stable allocation.

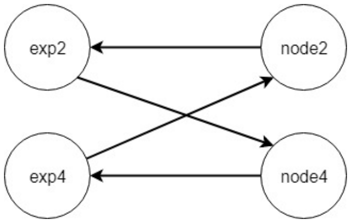


Fig. 2: TTC 2nd Iteration

In our case, nodes play the role of houses, while experimenters are essentially the owners. In more detail, the problem consists of a set of experimenters that provide a set of preferences over a finite amount of available nodes provided by the experimentation testbed. In order to better understand the applicability of the TTC algorithm over the node allocation problem, consider the following example where 4 experimenters provide their preferences over 4 available nodes as shown in Table I. Experimenters' first preferences are in the top row, while at the bottom the least preferred nodes can be found. For sake of simplicity, we consider for the initial allocation that i-th node belongs to i-th experimenter, meaning that node 1 belongs to experimenter 1 and so on.

In the first iteration that is presented in Figure 1, TTC will detect a cycle where experimenter 1 wants node 3, which belongs to experimenter 3 who in his turn wants node 1 that belongs to experimenter 1. A mutual exchange between these 2 experimenters is done and they get removed before the next iteration starts. In this iteration that can be seen in Figure 2, TTC detects a cycle between experimenter 2 and 4, thus an exchange is performed where experimenter 2 gets node 4 and experimenter 4 gets node 2. The final allocation of the TTC algorithm is shown in Table II, which is a core allocation, meaning that there is no other allocation an experimenter would have a greater preference for.

exp1	exp2	exp3	exp4
node 3	node 4	node 1	node 2

TABLE II: Final TTC Allocation

Combining the TTC algorithm together with the formulation of the problem presented in the previous section, we derived an extended TTC algorithm (See description in Algorithm 1). The algorithm takes into consideration the initial conditions where $\mathcal{A} = \{a_1, a_2, ..., a_M\}$ denotes the set of experimenters trying to reserve nodes, $\mathcal{H} = \{h_1, h_2, ..., h_N\}$ denotes the available nodes and $\mathbf{P} = \mathcal{P}_{a_1} \cup \mathcal{P}_{a_2} \cup ... \cup \mathcal{P}_{a_M}$ their preferences over these nodes. The algorithm continues as follows: In every iteration the TTC takes as input a random initial allocation, a set of the experimenters competing for the available nodes and their sets of preferred nodes. Each iteration results to a stable allocation that provides 1 node per experimenter. Before continuing, the allocated nodes are omitted from the set of available nodes, as well as from the preference sets of the experimenters. This can result in situations where the initial preferences of an experimenter cannot be served because requested nodes are no more available. In these situations, our algorithm randomly allocates a node from the remaining available nodes to the experimenter, in order to satisfy the cardinality of its initial preference set. The algorithm terminates when there are no available nodes left or after reaching $\max(|\mathcal{P}_{a_i}|)$ number of iterations.

For the sake of completeness, we describe in the context of our problem formulation, the FCFS mechanism in Algorithm 2. The algorithm terminates when there are no more experimenters or available nodes. In every iteration, the first experimenter gets all the nodes listed in its preference set $\mathcal{P}_{a_i}$ as it is the first experimenter using the system. Before moving to the next iteration, the allocated nodes are removed from the set of available nodes, as well as from the preference sets of the rest of the experimenters. In the situations where experimenters cannot receive the nodes, requested initially by their preferences, a random node from the available ones is allocated to them. This reflects the scenario where an experimenter had in his mind some resources to reserve but these resources were not available because someone else had already reserved them. In this case, he picks some resources from those that are available.

Algorithm 1 Top Trading Cycles (TTC)

```
$\mathcal{A} = \{a_1, a_2, ..., a_M\}$ denotes the set of experimenters trying to reserve nodes
$\mathcal{H} = \{h_1, h_2, ..., h_N\}$ denotes the set of available nodes
$\mathcal{P}_{a_i}$ denotes the preferred allocation for experimenter $a_i$
$\overline{\mathcal{P}}_{a_i}$ denotes the final allocation for experimenter $a_i$
while ($\mathcal{P}_a$ AND $\mathcal{H}$) are not empty do
    Initial random allocation of 1 node per experimenter $\rightarrow e$
    Run the TTC algorithm based on the market setup $\langle \mathcal{H}, \mathcal{A}, e \rangle$ (i.e. Detect cycles and mutual exchange opportunities)
    TTC results in a core allocation of 1 node per experimenter $\overline{\mathcal{P}}_{a_i}$
    In the case a node is no more available, a random node is allocated to the experimenter from the available ones
    Remove allocated nodes $\mathcal{H} = \mathcal{H} - \overline{\mathcal{P}}_a$
end while
```

Apparently the FCFS mechanism utilized in most experimentation testbeds, provides practical advantages as it is an easy to understand and implement scheme for managing the testbed resources, however it lacks fairness and efficiency. The FCFS scheme does not leave room for the testbed owner to improve the overall utilization of the testbed by allowing mutual beneficial exchanges of nodes between the experimenters.

Algorithm 2 First Come First Served (FCFS)

```
$\mathcal{A} = \{a_1, a_2, ..., a_M\}$ denotes the set of experimenters trying to reserve nodes
$\mathcal{H} = \{h_1, h_2, ..., h_N\}$ denotes the set of available nodes
$\mathbf{P} = \mathcal{P}_{a_1} \cup \mathcal{P}_{a_2} \cup ... \cup \mathcal{P}_{a_M}$ denotes the union of preference set by each experimenter
while ($\mathcal{A}$ AND $\mathcal{H}$) are not empty do
    Allocate the desired set of nodes $\mathcal{P}_{a_i}$ to experimenter $a_i$
    Remove allocated nodes from the set of available nodes $\mathcal{H} = \mathcal{H} - \mathcal{P}_{a_i}$
    Move to the next experimenter $i + 1$
end while
```

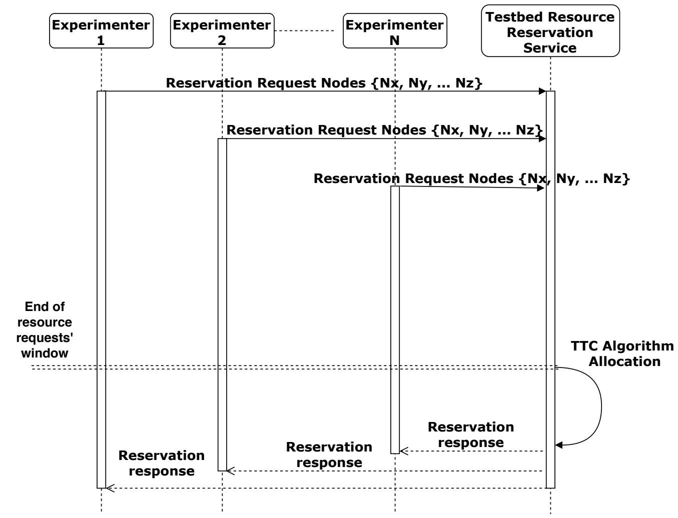


Fig. 3: TTC Interaction Diagram

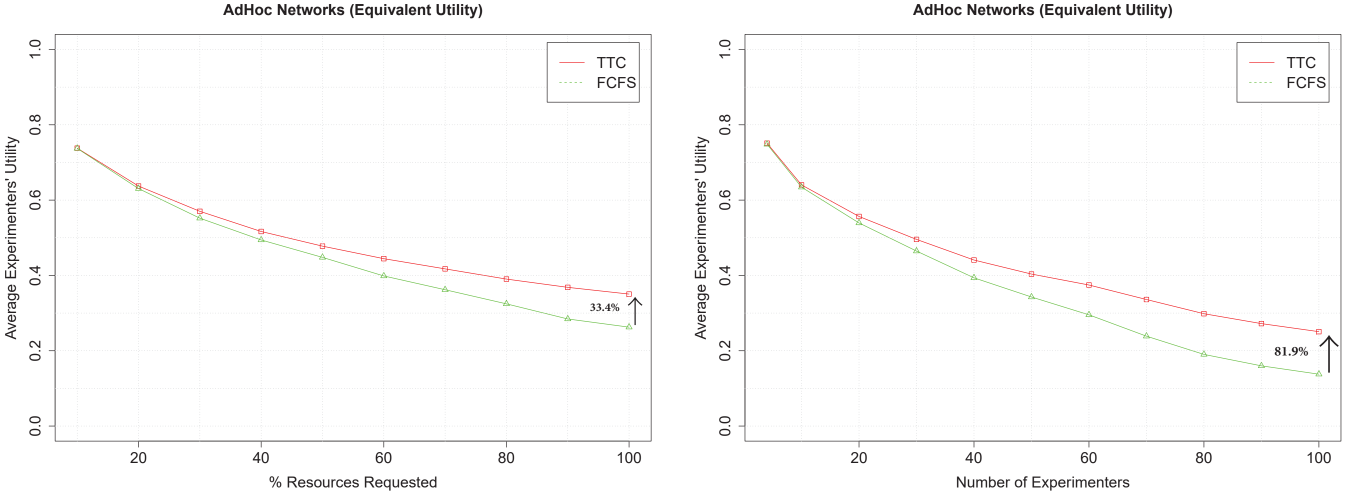
For the sake of providing an illustrative example of how TTC is collecting resource requests by the experimenters, before reaching a stable core allocation of resources based on the received requests, we present a flowchart in Figure 3 where the interaction of the experimenters with the testbed management services is depicted. More specifically, the experimenters would submit their requests with their resource preferences to the testbed reservation service up until a specific time which denotes the end of resource requests' window. At that time, the TTC would provide a Pareto efficient allocation based on the input received by the experimenters' preferences and the available nodes of the testbed. The difference with the FCFS algorithm is that experimenters receive a response to their reservation requests when the time window allowed for request submissions has ended. In the next section we provide an evaluation of the degree of improvement that TTC introduces compared to FCFS in terms of better utilization of the testbed's resources.

V. SYSTEM EVALUATION

A. Scenarios with Nodes with Equivalent Utility

In order to perform the evaluation of the proposed scheme, we conducted several simulations based on a set of scenarios that apply to our real-life experience from the daily operation of the NITOS experimentation testbed. More precisely, we started by testing the two algorithms described in the previous section with scenarios where the experimenters' preferences had equivalent utility over the nodes. We divided our simulations in 2 sets as follows:

- Gradually increase resource requests from 10% of total resources up to 100%.



(a) Increasing Requests

(b) Increasing Experimenters

Fig. 4: Requests for Nodes with Equivalent Utility

- Continuously increase number of experimenters, beyond the point of the 100% utilization.

For the first scenario of equivalent utility for all the nodes shown in Figure 4, we assumed a testbed with 400 total node resources with experimenter requests following a uniform distribution between 2 and 10 nodes, which reflects our experience with the case of the reservation requests of NITOS experimenters. The utility of the overall system is calculated based on equation (1). We can see in Figure 4a that as the requests of the experimenters increase reaching 100% utilization of the available resources, the TTC maintains the average normalised experimenter utility close to 0.35 while FCFS drops to 0.26. This is justified by the fact that an increased number of resource requests leads to more opportunities of mutual beneficial exchanges of resources, improving the utility of the system by 33.4% compared to FCFS approach in the case of 100% resource requests.

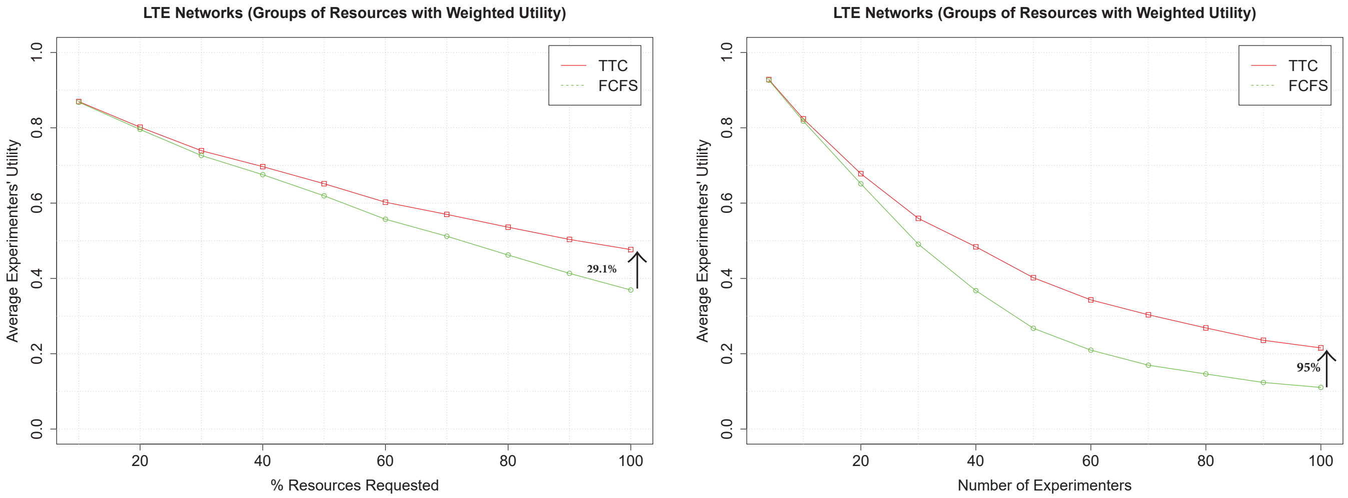
In our second set of simulations shown in Figure 4b, we can see how the system behaves when there is a continuously increasing number of experimenters leading to situations where the available resources are less than the requested. In this case, experimenters' requests will be rejected when the FCFS mechanism is in operation after all the resources have been allocated. The behavior of the TTC will be different as it collects all the preferences before deciding the final allocation and in this scenario tries to serve all the experimenters leading to less allocated nodes than the experimenters initially requested. This means that all the experimenters will experience the same amount of resource request rejections, leading to a more fair final allocation. Indicatively, we can see that when there are 100 experimenters trying to reserve resources, the TTC algorithm manages to provide 81.9% better utility than the FCFS algorithm.

B. Scenarios with Groups of Nodes with Weighted Utility

The above experiments apply when all the resources of a testbed are of equal importance to all experimenters and provide the same utility. However, this is far from accurate when different resources provide different utility for the experimenters, as in the 2nd scenario where we assumed that the resources are divided in 2 groups. The first one consists of LTE base stations (eNBs), while the second contains the LTE clients (UEs). The experimenters evaluate their preferences for eNBs higher than their preferences for UEs with utility weight 0.7 compared to 0.3. For this scenario we assumed that there are 40 eNBs and 400 UEs and each experimenter requests 1 eNB along with 10 UEs. The utility of the system is calculated by equation (5) where the different weights are taken into consideration for the overall utility.

Like in the 1st scenario we conducted 2 sets of simulations where there is an increasing number of resource requests in the 1st and an increasing number of experimenters in the 2nd. In Figure 5a we can observe similar behavior with the first scenario shown in Figure 4a with small differences in the total utility numbers, due to the fact that there are 2 groups of resources with different weights in the utility calculations. The first group, which contains the available eNBs for reservation is valued higher than the other group and since most of the experimenters get their preferred eNB, the average utility of the system is attained slightly higher compared to the previous scenario and improved by 29.1% compared to FCFS.

In the second set of simulations shown in Figure 5b, we can observe that the TTC algorithm continues to outperform the FCFS and achieves 95% better utility when there are 100 experimenters in the testbed. In this set of simulations, experimenters request a fixed amount of resources, which is 1 eNB and 10 UEs per experimenter. The TTC algorithm begins to perform more efficiently than the FCFS after the 20th experimenter, increasing the difference of the average system utility with the FCFS up to the maximum point of the 100 experimenters. Based on our findings, we can say that FCFS can only reach high average utility when there is no competition for reserving the same nodes as we notice for



(a) Increasing Requests

(b) Increasing Experimenters

Fig. 5: Requests for Groups of Nodes with Weighted Utility

low percentage at requested resources in Figure 5a and for lower number of experimenters in Figure 5b.

The values depicted in the graphs for the aforementioned scenarios were the outcome of extensive simulation runs. More specifically, every point on the graphs is an average of 1000 simulation runs, thus minimizing the effect of outliers and providing more accurate results. The TTC algorithm provides results instantaneously for most of the provided input (number of experimenters/resources). A realistic scenario from our day to day experience of testbed operation in NITOS testbed includes 100 experimenters and 500 resources, for which TTC can provide a stable allocation instantaneously. Even if we increase the number of experimenters and resources by an order of magnitude, which is 1000 experimenters and 5000 resources, the results can be obtained in less than 1 second. These performance results can be achieved by using commercial off-the-shelf hardware like desktop PCs. Hence, the TTC algorithm proves a feasible solution that can be easily adopted by the management and resource allocation services of the testbeds.

VI. CONCLUSION & FUTURE WORK

In this paper, we examined the problem of resource allocation in networking experimentation testbeds where resources cannot be shared among the experimenters and a reservation must be done prior to the provisioning of the resources. We highlighted the fact that a small number of studies have focused on the problem of efficient utilization of testbed resources mainly employing auction mechanisms to overcome the problem of resolving reservation conflicts for the same resources. We underline the similarities of the problem with that of the housing market and see the connections between resources/nodes and houses, as well as between experimenters and owners. We adopt the TTC algorithm which is proven to provide a Pareto efficient solution in exchange markets of indivisible goods without money. One of the key distinguishing characteristics of this approach is the lack of need for a closed economy and a virtual currency approach that introduces an overhead to the experimenters. With the approach that we present in this paper, the experimenters are only called to declare their preferences regarding resource reservation. In the performance evaluation we presented, the TTC algorithm clearly outperforms the current FCFS scheme used in most of the networking testbeds.

For future work, we plan to investigate scenarios where experimenters are getting rewarded if they decide to return reserved resources that for some reason are not utilized in their experiments or they are not needed for the whole reservation time period. This need comes from our experience of running a networking testbed where experimenters usually over-reserve resources in order to assess which ones they would use in the end. This results in under-utilization of the testbed and hinders other experimenters to use those over-reserved resources.

ACKNOWLEDGMENT

We acknowledge support of this work by the project Hellenic Research Infrastructure HELNET (MIS 5002781) which is implemented under the Action Reinforcement of the Research and Innovation Infrastructure, funded by the Operational Programme ”Competitiveness, Entrepreneurship and Innovation” (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

Fig. 6: Co-financed by Greece and the European Union

REFERENCES

[1] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, “Geni: A federated testbed for innovative network experiments,” *Computer Networks*, vol. 61, pp. 5–23, 2014.

[2] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, “Future internet research and experimentation: the fire initiative,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, pp. 89–92, 2007.

[3] 5G-PPP, “5G-PPP White Papers,” 2019, [Online], https://5g-ppp.eu/white-papers/.

[4] J. Van Ooteghem, S. Taylor, J. Ahtes, M. Smirnov, and M. Sawyer, “D2.10 final sustainability plan,” *deliverable of the FP7 Fed4FIRE project*, 2016.

[5] PAWR, “PAWR Program,” 2019, [Online], https://advancedwireless.org/.

[6] T. Wauters, B. Vermeulen, W. Vandenberghe, P. Demeester, S. Taylor, L. Baron, M. Smirnov, Y. Al-Hazmi, A. Willner, M. Sawyer *et al.*, “Federation of internet experimentation facilities: architecture and implementation,” in *European Conference on Networks and Communications (EuCNC 2014)*, 2014.

[7] NITOS, “Network Implementation Testbed using Open Source platforms.” 2019, [Online], http://nitlab.inf.uth.gr/NITlab/index.php/nitos.html.

[8] D. Stavropoulos, A. Dadoukis, T. Rakotoarivelo, M. Ott, T. Korakis, and L. Tassiulas, “Design, architecture and implementation of a resource discovery, reservation and provisioning framework for testbeds,” in *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2015, pp. 48–53.

[9] H. Niavis, K. Choumas, G. Iosifidis, T. Korakis, and L. Tassiulas, “Auction-based Scheduling of Wireless Testbed Resources,” in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, 2014.

[10] B. Chun, P. Buonadonna, A. AuYoung, C. Ng, D. Parkes, J. Shneidman, A. Snoeren, and A. Vahdat, “Mirage: A microeconomic resource allocation system for sensornet testbeds,” vol. 2005, pp. 19– 28.

[11] A. AuYoung, B. Chun, A. Snoeren, and A. Vahdat, “Resource allocation in federated distributed computing infrastructures,” in *Proceedings of the 1st Workshop on Operating System and Architectural Support for the On-demand IT InfraStructure*, vol. 9, 2004.

[12] M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding: substrate support for path splitting and migration,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[13] D. G. Andersen, “Theoretical approaches to node assignment,” Dec. 2002, Unpublished Manuscript, [Online], https://www.cs.cmu.edu/ dga/papers/andersen-assign-abstract.html.

[14] S. De Vries and R. V. Vohra, “Combinatorial auctions: A survey,” *INFORMS Journal on computing*, vol. 15, no. 3, pp. 284–309, 2003.

[15] L. Shapley and H. Scarf, “On cores and indivisibility,” *Journal of mathematical economics*, vol. 1, no. 1, pp. 23–37, 1974.

[16] R. W. Shephard and R. Färe, “The law of diminishing returns,” in *Production theory*. Springer, 1974, pp. 287–318.

[17] J. Stebbins, “The law of diminishing returns,” *Science*, vol. 99, no. 2571, pp. 267–271, 1944.

[18] P. Jaccard, “The distribution of the flora in the alpine zone. 1,” *New phytologist*, vol. 11, no. 2, pp. 37–50, 1912.