Cloud Federations: Economics, Games and Benefits

George Darzanos, Iordanis Koutsopoulos, *Senior Member, IEEE*, and George D. Stamoulis

***Abstract*— Sharing economy is a game-changing business paradigm that is currently permeating several industrial sectors. This paper aims to build a fundamental theory of the sharing economy of the computational capacity resource of Cloud Service Providers (CSPs). CSPs aim to cost-efficient serve geographically dispersed customers that often request computational resource-demanding services. The formation of CSP federations arises as an effective means to manage these diverse and time-varying service requests. In this paper, we introduce innovative federation models and policies for profitable federations that also achieve adequate QoS for their customers. Taking in account the flexible cloud computing service model, we abstract the virtualized infrastructure of each CSP to an M/M/1 queueing system, we formulate the CSP revenue and cost functions, and we study the task forwarding-based (TF) and the capacity sharing-based (CS) federation approaches. Under TF, each CSP may forward part of its workload to other federated CSPs, while under CS each CSP may share parts of its computational infrastructure with others. For both approaches, we propose two operation modes with different degree of CSPs' cooperation: (i) the joint business mode, where the CSPs fully cooperate: they jointly decide on the federation policies that maximize the total federation profit which is shared fairly among them; (ii) the reward-driven mode, where self-interested CSPs participate in a game: they adjust their responses to federation policies aiming to maximize their individual profits. The results reveal that our policies lead to effective federations, which are beneficial both for CSPs and for customers.**

***Index Terms*— Cloud federation, cloud economics, games, sharing economy, computational capacity, shapley value.**

I. Introduction

SHARING economy is a game-changing business paradigm that is currently permeating several industrial sectors, such as transportation (e.g. Uber, Lyft), lodging (e.g. Airbnb), and various online service exchange platforms. Resource federation and opportunistic resource sharing are the two main facets of what is recently known as the sharing economy. In resource sharing, the involved entities opportunistically share their resources by directly trading their excess supply for others' unsatisfied demand. On the other hand, in resource federation, sharing is achieved by aggregating the demand of many entities and by pooling their available resources. This paper aims to build a fundamental theory of the sharing economy of the computational capacity resource of Cloud Service Providers (CSPs).

Cloud computing promises ubiquitous, convenient and on-demand access to a shared pool of virtualized resources that can be rapidly provided and released with minimal management effort. Nowadays, the majority of application providers are moving to the cloud, since they can take advantage of its flexible and scalable resources and thus avoid investments on costly hardware infrastructures. However, the broad range of cloud-based applications today (VoD, on-line gaming, etc.) exhibit high temporal variations of their workload, and require high QoS (Quality of Service) for end-users that may be dispersed in multiple geographical areas. In order to satisfy the increasingly demanding performance requirements of these applications, CSPs could invest on building additional infrastructures so as to achieve service coverage in all geographical locations where significant demand is observed; however, even market giants do not find such an approach profitable [1]. Cloud federation is emerging as an effective solution for expanding CSP's geographic presence and cost-effective servicing of customer requests in a manner that respects QoS requirements. In a cloud federation, multiple CSPs cooperate to seamlessly provide services to customers that reside in the administrative domain of different CSPs.

Examples of federated clouds: Today, several academic or commercial platforms already enable the realization of cloud federations in practice. The OnApp Federation [2] is a network of Infrastructure as a Service (IaaS) that connects multiple CSPs running the OnApp cloud management platform. The CSPs interact through the OnApp market, where they can sell or buy computational resources on demand. Further, Arjuna's Agility framework [3] is a dynamic pool of IT resources that are offered by different administrative domains within one or multiple enterprises. EGI Federated Cloud [4] is a seamless grid of academic private clouds and virtualized resources, which is built according to open standards and focuses on the computational requirements of the scientific community. BonFIRE [5] offers a federated testbed that supports large-scale testing of applications, services and systems over multiple, geographically dispersed, heterogeneous cloud and network testbeds. Finally, the CERN Openlab project [6] aims to build a seamless federation among multiple private and public cloud platforms on OpenStack.

Incentives for Cloud Federation: Cloud federation incentivizes CSPs to participate due to potential CAPEX and OPEX

Manuscript received December 18, 2017; revised February 20, 2019; accepted August 26, 2019; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Shakkottai. Date of publication October 8, 2019; date of current version October 15, 2019. This work was supported in part by the European Commission through the Framework of the H2020-ICT-2014 project 5GEx. This information reflects the consortium's view, but neither the consortium nor the European Commission are liable for any use that may be done of the information contained therein. This work was also supported in part by the Research Centre of AUEB through the Framework of the internal project EP-2791-01: Original Scientific Publications 2017–18. The work of I. Koutsopoulos was supported by the internal AUEB project ER-3037-07 Original Scientific Publications 2018–2019. *(Corresponding author: George Darzanos.)*

The authors are with the Department of Informatics, Athens University of Economics and Business, 10434 Athens, Greece (e-mail: ntarzanos@aueb.gr; jordan@aueb.gr; gstamoul@aueb.gr).

Digital Object Identifier 10.1109/TNET.2019.2943810

reduction. CAPEX can be reduced since CSPs do not have to over-dimension their infrastructures to serve the peaks of demand, while OPEX (e.g. energy cost) can be lessened by means of inter-CSP load balancing. In order to achieve the above benefits, a cloud federation should comprise a set of policies and rules that guarantee the high QoS as well as the economic sustainability both of the federation as a whole and of its participants individually. Thus, each CSP that joins a federation should comply with associated policies and rules that define how CSPs interact and cooperate in order to jointly provide services and share profits.

Challenges: Central in cloud federations is the issue of *resource allocation and coordination*, since CSPs have diverse resources, and they serve customers with diverse needs in different locations. The prime types of resources considered in CSP federation are computational capacity (cycles) and storage. In this work, we consider the coordination of allocation of computational capacity among federated CSPs. The reason is that computational capacity is far more challenging than storage. First and foremost, a unit of computational capacity is much more expensive than a unit of storage capacity, thus it makes much more sense to focus on this resource. Second, the dynamics of computational cycle allocation are more involved due to time-varying workloads. Third, most of the emerging applications involve computationally intensive tasks that take significantly more time compared to other aspects of such applications. There are several works in the recent literature that look at resource allocation and deal with *cooperative resource pooling* [7]–[9], where CSPs aggregate their computational resources in a virtualized infrastructure that serves the needs of all CSPs' customers, or *resource trading* [10]–[13], where the CSPs make profit from contributing their idle resources to server customers of other federated CSPs.

A. Our Contribution

In this paper, we build on and substantially extend our prelude work [14], [15]. We model each CSP as an $M/M/1$ queue that also provides guarantee regarding the worst QoS its customers may experience, and we provide mathematical models for the CSP revenue and cost. In particular, we assume that the cost of a CSP is due to the energy consumed by its infrastructure, while its revenue arises from a QoS-dependent pricing on its customers. We introduce two alternative cloud federation approaches: (i) The task forwarding-based (TF) cloud federation, in which the portion of request workload to be transferred from each CSP to other federation participants is decided (*TF policy*). (ii) The capacity sharing-based (CS) cloud federation, in which the part of computational resources of each CSP to be granted to other federated CSPs is decided (*CS policy*).

For both TF and CS, we propose and analyze economic policies for two different modes, namely the joint business and the reward-driven federation modes. In a *joint business federation*, the participating CSPs cooperate to serve the aggregate federation workload with the objective to maximize their total profits. Then, these profits are shared among them based on some rule agreed a priori in such a way that each individual CSP benefits from the formation of the federation. We propose the use of the state-of-art Shapley value notion (see [16]) as an effective approach for the fair distribution of total profit. Due to the high complexity of this approach as the number of CSPs increases, we also introduce an alternative heuristic approach that scales nicely without introducing additional complexity and achieves comparable results.

In a *reward-driven federation*, the CSPs are again collaborative with respect to service provisioning, however each of them selfishly determines the level of its contribution to the federation by adjusting its own TF or CS policy. Each CSP aims to maximize its individual profit; thus, a non-cooperative game arises. However, we design this game in a way that the payoff function of each CSP includes a reward mechanism that is selected appropriately to incentivize the self-interested profit-seeking CSPs to contribute to the federation. In particular, we take that the Shapley value of each CSP is agreed to be its payoff in this game. One of our main findings is that reward-driven federation converges to a unique equilibrium that is equivalent in terms of profits to a joint business federation.

The rest of this paper is organized as follows. In section II, we present the model for one CSP. In section III, we present the TF and CS federation approaches. For these approaches, in section IV we define economic policies for the two different federation modes. In section V we present our numerical evaluation results. In section VI we overview the relevant state-of-the art work, and in section VII we present our conclusions.

II. Cloud Service Provider Model

A real-world public CSPs consists of multiple datacenters with a number of physical hosts in each of them. The virtualization technology of cloud computing allows both pooling and slicing of the physical resources, thus enabling the flexible resource provisioning and multi-tenancy over the physical hosts. Specifically, each CSP offers public access to virtualized computational and storage resources through the Internet in the form of Virtual Machines (VMs) or Containers. Examples of such IaaS platforms are the Amazon EC2, Google Compute Engine, Microsoft Azure and IBM Cloud.

In our approach, we assume that customer-generated task requests arrive at each CSP in the form of a stream. Each request corresponds to single "task" regardless if this task includes multiple jobs. The CSP that receives a request translates it into one or more VMs able to fulfill the respective service. However, VMs are not in abundance, but they are finite resources that are assigned on-demand to serve requests. The flexible cloud computing service model can be effectively captured by means of queueing theory. In particular, if we assume that the whole infrastructure (i.e. physical servers) of a CSP is virtualized and pooled by means of a central Resource Orchestrator (RO), it can then be sliced into c identical VMs, each of them having adequate computational capacity to serve a task request to the required QoS. By intuition, this leads us to the adoption of an $M/M/c$ queueing system, where all customers' requests arrive to a single queue, each to be served by one of the c identical VMs. However, in our work, we abstract this multi-server model to a single-server $M/M/1$ queueing system, where the whole CSP infrastructure is pooled into one omni-powerful server that is fully utilized when

serving one-by-one the incoming service requests. While this assumption should be understood as a modeling convention that allows better mathematical treatment of our paper, it is reasonable enough to capture reality and it is justified and supported through a discussion in subsection II-A and our numerical results in subsection V-B.1.

A. A CSP as an M/M/1 Queue

We abstract the service model of a CSP to a single-server $M/M/1$ queueing system. In particular, assuming that the infrastructure of a CSP i consists of M_i identical physical servers of computational capacity C_i/M_i each, we assume that the CSP performs pooling of these resources into a unified virtual infrastructure of total capacity C_i. Hence, we assume that the multi-physical server infrastructure of each CSP behaves as a *single-server* system with computational capacity C_i and utilization ρ_i.

We assume that service requests from the customers of CSP i arrive to its RO according to a Poisson process of rate λ_i (tasks/sec). Each of these tasks requires a random number of operations in order to be executed. We assume that the number of operations follows an exponential distribution with mean number L operations/task. Thus, the average service rate (in tasks/sec) for a CSP i is $\mu_i = C_i/L$, and therefore the service time of a task is exponentially distributed with mean $1/\mu_i$. We assume that the task requests are not impatient, the probability to be withdrawn before served is assumed to be zero.

We use the *average task completion time*, i.e. the task waiting time in the queue and its service time, as a proxy for customers' QoS. By standard queueing theory, the average completion time $d_i(\cdot)$ for tasks served by the infrastructure of CSP i is given by $d_i(\lambda_i) = \frac{1}{\mu_i - \lambda_i}$. Note that the average rate of incoming tasks must always be lower than the service rate of the system ($\lambda_i < \mu_i$), otherwise the CSP queue becomes unstable. Given that d_i is increasing and convex in λ_i, as λ_i approaches μ_i, the value of d_i increases without upper bound, which is an undesirable situation for the customers. A CSP that wishes to keep its quality at least acceptable should avoid a very high utilization ($\rho_i = \frac{\lambda_i}{\mu_i}$) of its infrastructure. To this end, we assume that each CSP takes into account its customers' SLAs and determines which is the worst tolerable QoS level (e.g. the maximum average task completion time, d_i^{max}). Based on this QoS level, CSP i derives the maximum acceptable utilization level of its infrastructure.

Next, we discuss and justify the main assumptions with respect to the $M/M/1$ modeling:

- ***Poisson arrivals.*** Given that a typical cloud computing system serves a large number of customers where each of them may generate multiple computational tasks, is more probable to have bursts of task requests with smaller inter-arrival times than larger ones. Thereafter, we can assume that the tasks arrive according to a Poisson process.
- ***Exponential service time.*** The time that a task spends in the CSP's system depends both on the waiting and service time, i.e. on the number of existing tasks that wait to be served, on the availability of resources when the task arrives and on its size with respect to the number of operations it entails. The majority of tasks that arrive in a CSP queue usually demands a smaller number of operation, while relatively fewer tasks require a large number of operation. Hence, we assume that the number of operations that a task requires is exponentially distributed, and therefore its service time also follows an exponential distribution.
- ***Single server abstraction.*** We argue that abstracting the infrastructure of a CSP i to an $M/M/1$ queueing system of capacity C_i, instead of an $M/M/c$ system with c servers of capacity C_i/c each, does not affect the qualitative properties of the results of our analysis. This happens because both models exhibit similar behavior on the characteristics we are interested in. First and foremost, the average task completion time in both $M/M/1$ and $M/M/c$ modeling approaches is a convex and increasing function of the task arrival rate λ_i. This observation, combined with the fact that in our model we are only interested in the *average* completion time of a task that enters our system, renders both $M/M/c$ and $M/M/1$ modeling approaches applicable. Further details on the actual queueing that takes place in the system and the performance of individual tasks does not affect the behavior of the proposed models and policies, and thus are not of our interest. Finally, the performance of an $M/M/c$ system approaches the one of $M/M/1$ under heavy loads becoming equal when $\rho_i \cong 1$.

In section V-B.1 we present numerical results demonstrating that the use of the $M/M/c$ model would have not affected the qualitative outcomes of our analysis. Therefore, we proceed with employing the $M/M/1$ model.

B. CSP Cost and Revenue

Energy Consumption Cost per Time Unit: The operational expenses of a CSP include costs related to the power consumption of its infrastructure, and the floor space, storage and IT operations to manage this infrastructure. These costs can be considered as fixed in a given time period except for the power consumption of the infrastructure since it depends on the CSP workload. Hence, we assume that the operational expenses a CSP come primarily from energy consumption of its infrastructure.

Given that the power consumption of a server includes the power for its operation and that required for supportive systems like cooling devices, the operational cost of a CSP is determined by the utilization of its servers. The total power consumption is a linearly increasing function of the utilization factor of the server, ρ [17]. Specifically, the total consumed power is the sum of server's idle power and utilization factor-dependent dynamic power consumption. The former one, W_0, is the power consumed when the server is powered on but does not serve any tasks. The latter one is linearly increasing in the server utilization ρ. If W_1 is the total power of a server when it is fully utilized (namely at $\rho = 1$), then the dynamic power consumption ranges from 0 to $W_1 - W_0$.

Considering that a CSP maintains multiple physical servers up-and-running to handle the incoming demand at any given

time, the central RO performs a perfect slicing of the virtualized pool of resources in order to evenly distribute the load among physical servers. Thus, we reasonably assume that all CSP servers operate at the same utilization level ρ.

To this end, the idle and dynamic power consumptions of the entire infrastructure can be computed by aggregating the corresponding power consumption patterns of all servers. Then, if a CSP i has M_i servers, and if $W_{0,ij}$ and $W_{1,ij}$ denote the idle and total power consumption of the j-th server of CSP i, then the aggregate power consumption of the CSP in Watts is

$$W_i(\lambda_i) = W_{0,i} + \Big(W_{1,i} - W_{0,i}\Big)\frac{\lambda_i}{\mu_i}, \tag{1}$$

where $W_{0,i} = \sum_{j=1}^{M_i} W_{0,ij}$ and $W_{1,i} = \sum_{j=1}^{M_i} W_{1,ij}$ denote the total idle and total power consumptions of i's infrastructure. If CSP i is supplied electricity at a price z_i per KWatt·sec, then the cost of energy consumption per unit of time (cash outflow) is given by $E_i(\lambda_i) = W_i(\lambda_i)z_i$.

Note that formula (1) could be broken down into the sum of all individual servers' power consumption (idle/dynamic) in case of non-identical servers or not evenly distribute loads among them.

QoS-Dependent Pricing: We assume that a CSP charges its customers based on the offered QoS level and on the number of received requests per customer. Recall that we use the average task completion time (d_i, also referred to henceforth as delay) as a measure for the QoS offered by a CSP. A CSP i sets a price per task according to a pricing function $p_i(\cdot)$, where $p_i(\cdot)$ is decreasing in d_i. Further, d_i is lower-bounded by the expected service time $d_i^{min} = 1/\mu_i$ which corresponds to the delay when no queueing of tasks occurs. Recall that d_i^{max} is the worst QoS that a customer can tolerate. A pricing function that satisfies the requirements above is

$$p_i(\lambda_i) = \Big(1 - \frac{d_i(\lambda_i) - d_i^{min}}{d_i^{max}}\Big)q_i, \tag{2}$$

where q_i denotes the price per task that i charges for offering service in the best possible QoS, i.e. d_i^{min}. At the end of this section we provide a discussion on pricing and potential alternative function.

In practice, the pricing policy of each CSP is partly driven by the competition in the market. In our approach, we assume that each CSP has made a decision offline on its pricing function that already takes competition into account. Moreover, we assume that CSPs cannot adapt their pricing functions and also that their customers are committed by some contract and therefore cannot change their serving CSP.

Revenue per Time Unit: The revenue of a CSP is generated from charging tasks. Since CSP is committed to serve tasks and there is no limit on the queue size, the tasks arrive in its queue are always completely served. Recall that we assume that the tasks are not impatient, thus the probability to be withdrawn before served is zero. Consequently, the revenue rate (cash inflow) in monetary units per unit of time for CSP i is given by

$$R_i(\lambda_i) = \lambda_i p_i(\lambda_i). \tag{3}$$

Net Profit per Time Unit: The profit (net cash inflow) that i earns per time unit is

$$P_i(\lambda_i) = R_i(\lambda_i) - E_i(\lambda_i). \tag{4}$$

Discussion I: QoS is an integral part of pricing in cloud computing since a CSP takes into account the characteristics (computation, memory, etc.) of a VM instance to determine its price per time unit. Accordingly, our pricing function (2) determines the price based on the customers' average delay. Note that alternative QoS-depended pricing functions could be used. For instance, p_i could be a convex function to the average task completion time in order to capture that a marginal change in delay is perceived more by the customers for smaller values of delay. Such functions are $p_i(\lambda_i) = \frac{d_i^{min}}{d_i(\lambda_i)}q_i$ and $p_i(\lambda_i) = \frac{1}{e^{d_i(\lambda_i) - d_i^{min}}}q_i$. In general, there is broad variety of functions that could be applied, however the characteristics of the pricing function directly affects the properties of CSP's revenue function (3). Hence, when it comes to the formation of federations that aims for profit maximization, pricing may have an impact on the CSPs' behavior. How the characteristics of a pricing function affects the behavior of CSPs under our federation models is discussed in Remark I in section III-B.

III. Two Federation Approaches

In this section we present the task forwarding-based and capacity sharing-based federation approaches. These two approaches represent the most important and common real-life scenarios, wherein a CSP may carry out computational tasks of customers of other CSPs or may share parts (slices) of its spare computational resources (eg. at times of low demand) to other CSPs to use it as if they were their own resources. Although both TF and CS address the usage of computational capacity of a CSP to serve the needs of other CSPs, they differ substantially in the mathematical formulation, as well as in the derived optimal policies and properties of equilibrium.

A. Task Forwarding-Based Federation Approach

In TF approach, each CSP may forward portions of the tasks coming from its own customers (incoming task stream) to other CSPs within the federation. The *TF policy* of each CSP determines the portion of its incoming task stream to be executed locally by its own infrastructure and the portions to be forwarded to each of the other CSPs.

We consider a set $\mathcal{N}$ of $N = |\mathcal{N}|$ CSPs, and for each CSP $i \in \mathcal{N}$ we define the probabilities α_{ij} for $j = 1, .., N$ ($\sum_j \alpha_{ij} \leq 1$) that determine the probability for a task of CSP i to be forwarded to a CSP j. As a consequence, these probabilities determine the portion of its incoming tasks stream that CSP i forwards to CSP j. The collective TF policy of all CSPs forms an $N \times N$ dimensional matrix $\mathbf{A}$, with entries α_{ij}. We use vectors $\mathbf{a}_i$ and $\mathbf{a}'_i$ to denote the i-th row and i-th column of $\mathbf{A}$ respectively. The total rate of task streams that

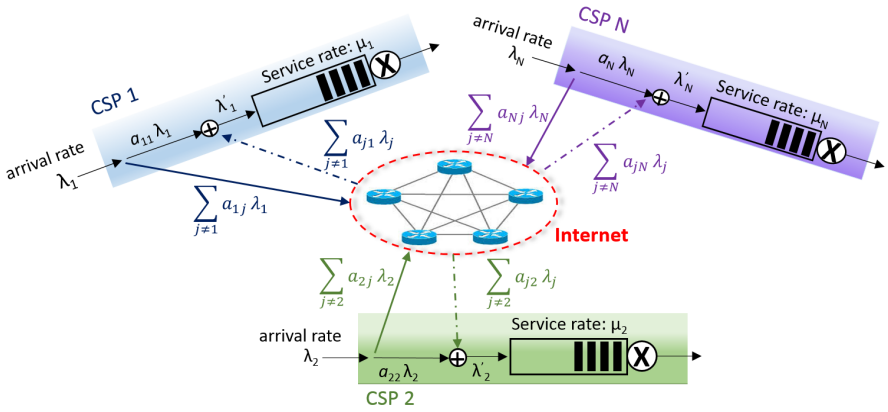


Fig. 1. TF approach for N CSPs, each of them is modeled as a $M/M/1$ queue. Each CSP may forward a portion of the task stream coming from its customers to others and likewise it can receive task streams coming from customers of others. The forwarded tasks undergo a fixed average transfer delay.

CSP i forwards to others is $\sum_{j\in\mathcal{N}\setminus\{i\}} \alpha_{ij}\lambda_i$, while the aggregate rate of task streams that CSP i receives from other CSPs is $\sum_{j\in\mathcal{N}\setminus\{i\}} \alpha_{ji}\lambda_j$. Fig. 1 depicts TF approach for N CSPs. We assume that the tasks transferred from a CSP to another experience an additional communication delay due to the intermediate Internet links between corresponding datacenters and possibly other factors associated with such a migration. For each pair of CSPs i, j we define an a priory to resource allocation average communication delay D_{ij}. Note that $D_{ii} = 0$. As this communication delay increases, it deters CSPs from outsourcing tasks since it becomes more beneficial to operate the local resources in a higher utilization level than outsourcing tasks to a remote CSP with a lower utilization level. The impact of communication delay is studied in section V-B.6.

The task request arrival rate at the input of each CSP queue now depends on the TF policies of all CSPs, namely on the i-th column of matrix $\mathbf{A}$ (i.e. on vector $\mathbf{a}'_i$), and it is defined as $\lambda'_i(\mathbf{a}'_i) = \sum_{j\in\mathcal{N}} \alpha_{ji}\lambda_j$. Note that α_{ii} determines which portion of the incoming task stream of i will be processed locally by i's infrastructure. The average completion time of tasks that are served by the infrastructure of CSP i is now given by

$$d_i(\mathbf{a}'_i) = \frac{1}{\mu_i - \lambda'_i(\mathbf{a}'_i)}. \tag{5}$$

A portion of the stream of tasks coming from customers of CSP i may be served by other CSPs, thus the average task completion time may depend on the average delay experienced at multiple CSP queues. To this end, the average task completion time for customers of CSP i depends on all columns of matrix $\mathbf{A}$ and is defined as

$$T_i(\mathbf{A}) = \sum_{j\in\mathcal{N}} \alpha_{ij}\big(d_j(\mathbf{a}'_i) + D_{ij}\big). \tag{6}$$

It is important to stress the difference between $T_i(\cdot)$ and $d_i(\cdot)$. $d_i(\cdot)$ is the average completion time for tasks that are served by the infrastructure of CSP i, including those tasks generated by customers of i and tasks from other CSPs' customers. $T_i(\cdot)$ is the average completion time of tasks that are generated by customers of CSP i, regardless of whether they are ultimately served by CSP i or by other CSPs.

In Section II, a complete characterization of a single CSP was provided, however here we slightly augment our model to fit the federation setup. The power consumption of CSP i's infrastructure is also affected by the TF policies of other CSPs and is given by

$$W_i(\mathbf{a}'_i) = W_{0,i} + \big(W_{1,i} - W_{0,i}\big)\frac{\lambda'_i(\mathbf{a}'_i)}{\mu_i}. \tag{7}$$

Accordingly, the energy cost per unit of time is defined as $E_i(\mathbf{a}'_i) = W_i(\mathbf{a}'_i)z_i$. Furthermore, the customers of CSP i should be charged based on $T_i(\cdot)$ rather than $d_i(\cdot)$ because part of these tasks may be served from different CSP queues. Hence, the pricing function becomes

$$p_i(\mathbf{A}) = \Big(1 - \frac{T_i(\mathbf{A}) - d_i^{min}}{d_i^{max}}\Big)q_i, \tag{8}$$

while the revenue and profit per unit of time become $R_i(\mathbf{A}) = \lambda_i p_i(\mathbf{A})$ and $P_i(\mathbf{A}) = R_i(\mathbf{A}) - E_i(\mathbf{a}'_i)$.

B. Capacity Sharing-Based Federation Approach

The CS approach relies on the computational resource sharing among the federated CSPs, i.e. a CSP can grant part of its computational capacity to others in the form of *resource slices*. For each CSP $i \in \mathcal{N}$ we define probabilities β_{ij} for $j = 1, .., N$ ($\sum_j \beta_{ij} \leq 1$) that determine its *CS policy*. For example, assuming that CSP i owns infrastructure of capacity C_i, $\beta_{ij}C_i$ is the computational capacity that CSP i grants to CSP j. Note that β_{ii} determines the part of local infrastructure owned by CSP i which remains under the control of i. The CS policies of all CSPs form an $N \times N$ dimensional matrix $\mathbf{B}$. Again, vectors $\mathbf{b}_i$ and $\mathbf{b}'_i$ denote the i-th row and i-th column of $\mathbf{B}$ respectively.

Under the CS approach, a CSP may have control over multiple resource slices located on different CSPs. Again, we assume that the RO of each CSP pools all these slices, thus each CSP can be still considered as an $M/M/1$ queue. To this end, the computational capacity of a CSP i depends on the CS policies of all CSPs and its value is given by aggregating the capacities of all resources that are pooled under its RO,

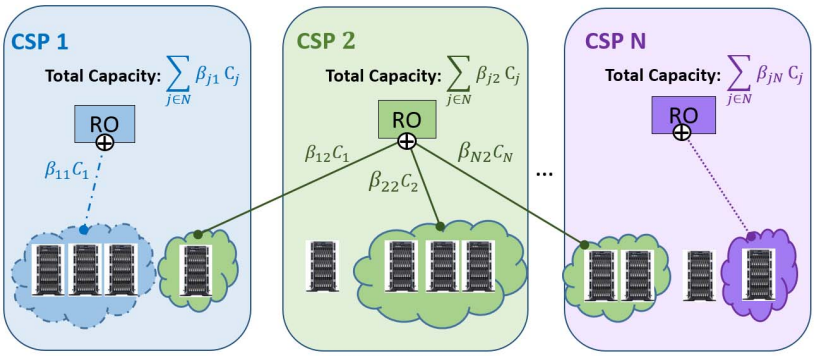


Fig. 2. The CS approach for N CSPs, where a CSP is granted parts (slices) of other CSPs' infrastructure to use them as its own. The RO of each CSP aggregates the resources under its control into a virtual pool which can be modeled as a single-server $M/M/1$ queue.

namely $C'_i(\mathbf{b}'_i) = \sum_{j\in N} \beta_{ji} C_j$, while the service rate of CSP i is given by

$$\mu'_i(\mathbf{b}'_i) = \frac{C'_i(\mathbf{b}'_i)}{L_i}. \tag{9}$$

Fig. 2 depicts the CS approach for N CSPs.

Given that we again abstract the infrastructure controlled by a CSP to an $M/M/1$ system, we should consider that the whole infrastructure as a unique system that serves all the CSP's workload. Hence, the task requests are proportionally distributed in all parts of the controlled infrastructure. This means that the portion of tasks requests being outsourced to each of the remote resource slices is $\frac{\sum_{j\in N\setminus\{i\}} \beta_{ji} C_j}{C'_i(\mathbf{b}'_i)}$. Consequently, the average completion time of tasks served by the infrastructure controlled by CSP i is given by

$$\tilde{T}_i(\mathbf{b}'_i) = \frac{1}{\mu'_i(\mathbf{b}'_i) - \lambda_i} + \frac{\sum_{j\in N\setminus\{i\}} \beta_{ji} C_j}{C'_i(\mathbf{b}'_i)} D_{ij} \tag{10}$$

where the first term in the formula above is the average task completion time given by standard $M/M/1$ queueing model, while the second term captures the communication cost for tasks transfered over the Internet to remote infrastructure that is controlled by CSP i, but it is owned by others. If the inter-CSP delay is neglected (i.e. $D_{ij} = 0$, $\forall i, j \in \mathcal{N}$), then the second term is omitted.

The different slices of infrastructure owned by a CSP may be utilized and controlled by others. For instance, in Fig. 2, the utilization of $\beta_{12}C_1$ slice is determined by the workload of CSP 2, while the utilization of $\beta_{11}C_1$ slice is determined by the workload of CSP 1. Thus, the average utilization of the infrastructure *owned* by CSP i is given by

$$\tilde{\rho}_i(\mathbf{B}) = \sum_{j\in N} \beta_{ij} \frac{\lambda_j}{\mu'_j(\mathbf{b}'_j)}. \tag{11}$$

We assume that the energy consumption cost of the resources owned by a CSP is payed by the owner CSP, even if these resources are utilized by others. Hence, this cost is included in the compensation mechanism. The power consumption of the infrastructure *owned* by a CSP i is affected by the average utilization of the different slices of its infrastructure $\tilde{\rho}_i(\mathbf{B})$ and thus it is a function of CS policy $\mathbf{B}$, $\tilde{W}_i(\mathbf{B}) = W_{0,i} + (W_{1,i} - W_{0,i})\tilde{\rho}_i(\mathbf{B})$. To this end, the energy consumption of infrastructure owned by i is now given by $\tilde{E}_i(\mathbf{B}) = \tilde{W}_i(\mathbf{B}) z_i$.

As in TF approach, the customers of CSP i should be charged based on $\tilde{T}_i(\cdot)$, hence the QoS-dependent pricing function becomes

$$\tilde{p}_i(\mathbf{b}'_i) = 1 - \frac{\tilde{T}_i(\mathbf{b}'_i) - d_i^{min}}{d_i^{max}} q_i \tag{12}$$

while the revenue is given by $\tilde{R}_i(\mathbf{b}'_i) = \lambda_i \tilde{p}_i(\mathbf{b}'_i)$. The profit of CSP i per unit of time is given by $\tilde{P}_i(\mathbf{B}) = \tilde{R}_i(\mathbf{b}'_i) - \tilde{E}_i(\mathbf{B})$.

Remark I. In CS federation, the capacity controlled by a CSP can be expanded beyond its own C_i capabilities by pooling resource slices granted from others. Hence, due to the $M/M/1$ modeling the best possible QoS that a CSP can provide to its customers can now become better than d_i^{min}. Hence, the customer may have to actually pay a higher price than q_i. This observation combined with the adoption of pricing functions that have certain characteristics may lead the profit-oriented CS federation to an "unfair" state for the customers. In fact, if the selected pricing function renders the CSP revenues a convex function to the service rate $\mu'_i(\mathbf{b}'_i)$, then the CS federation may be led to a state where the QoS for a large portion of a customers is extremely high, while the rest experience a QoS close to the worst acceptable. On the other hand, a pricing function that renders the CSP revenues *non-convex* to the service rate $\mu'_i(\mathbf{b}'_i)$ achieves load balancing and a state where all customers experience a QoS close to the federation's average. Such types of "unfair" pricing functions and potential ways to mitigate their impact are studied in subsection V-B.8.

IV. Federation Modes

For both federation approaches of section III, we propose different modes under which a federation can be formed, namely the (i) the *joint business* and the (ii) *reward-driven* federation modes. The two modes differ in the level of cooperation among CSPs who may have common or conflicting objectives, and in the type of private information that each CSP makes available to others.

A. Joint Business Federation

This mode is suitable for federations of fully cooperative CSPs. The CSPs that participate in a joint business federation comply to certain cooperation rules that have been agreed a priori. These rules include: (i) technical alignment of their infrastructure (ii) agreement to share key private information, e.g. the values of their computational capacity C_i and average request load λ_i, (iii) agreement on the common objective of total federation profit maximization, and (iv) cooperation in defining the appropriate policy for sharing the total profit incurred from the federation. Next, we present the joint business federation for the TF and CS approaches.

1) Task Forwarding-Based Joint Business Federation: The CSPs that form an TF joint business federation cooperate and *jointly decide the collective TF policy* $\mathbf{A}^*$ that maximizes the total profit of the federation. This globally optimal TF policy is derived by solving the following maximization problem,

$$\begin{aligned} \arg\max_{\mathbf{A}} \ & \sum_{i\in\mathcal{N}} P_i(\mathbf{A}) \\ \text{s.t. } & \alpha_{ij} \geq 0, \quad \forall i,j \in \mathcal{N}, \\ & \sum_{j\in\mathcal{N}} \alpha_{ij} = 1, \quad \forall i \in \mathcal{N}, \\ & T_i(\mathbf{A}) \leq d_i^{max}, \quad \forall i \in \mathcal{N}. \end{aligned} \tag{13}$$

The first two constraints are related to the splitting of CSP i's stream of task requests across all CSPs, while the third constraints guarantees that the achieved QoS is better than the worst acceptable d_i^{max}. We can solve this non-linear problem by applying standard optimization methods, i.e. formation of the Lagrangian and statement of the necessary and sufficient KKT conditions that should be satisfied for optimality.

Our problem formulation guarantees that under the optimal $\mathbf{A}^*$ the total federation profit is maximized. Also, an important property is that, under the optimal policy $\mathbf{A}^*$, a CSP will either forward or receive requests (never both). This has been proven analytically and is demonstrated through numerical results in section V. In the worst case scenario, i.e. $\mathbf{A}^* = \mathbf{I}$ (Identity matrix), the total federation profit equals the aggregate profit of CSPs in standalone operation. By standalone, we mean that each CSP serves only the tasks coming from its own customers. The third constraint of the maximization problem guarantees the offered QoS of a CSP does not drop below the worst acceptable, however the individual profit may in fact deteriorate for one (or more) CSPs due to task forwarding actions. Specifically, a CSP that only receives forwarded tasks by others experiences loss because the extra workload will downgrade the QoS of the CSP which leads to reduction in revenues, while it will also increase its energy cost due to the higher infrastructure utilization. (such a case always arises for $N = 2$.) As a result, such CSPs will be unwilling to comply with the federation, unless some rule is applied for their compensation. Since the total profit of the federation exceeds the aggregate profit of CSPs in the standalone mode, CSPs that only forward tasks definitely have higher profit than in standalone, thus they are able to compensate others.

2) Capacity Sharing-Based Joint Business Federation: All CSPs cooperatively decide on a collective CS policy $\mathbf{B}^*$ that maximizes the total profit of the federation by solving the respective maximization problem,

$$\begin{aligned} \arg\max_{\mathbf{B}} \ & \sum_{i\in\mathcal{N}} P_i(\mathbf{B}) \\ \text{s.t. } & \beta_{ij} \geq 0, \quad \forall i,j \in \mathcal{N}, \\ & \sum_{j\in\mathcal{N}} \beta_{ij} = 1, \quad \forall i \in \mathcal{N}, \\ & \tilde{T}_i(\mathbf{b}_i') \leq d_i^{max}, \quad \forall i \in \mathcal{N}. \end{aligned} \tag{14}$$

Again, the first two constrains capture the partitioning and sharing of resources to different CSPs while the third one guarantees that the bound for the worst acceptable QoS is not violated. Same as in TF approach, by solving this problem we obtain the CS policies $\mathbf{B}^*$ that maximize the total profit of the federation, but there is no guarantee regarding the profit of each individual CSP. To this end, we again have to define fair profit sharing policies in order to ensure that non CSP will incur losses. Appropriate such profit-sharing rules are discussed below.

3) Profit Sharing Rules: As explained, profit sharing rules should be applied on the total profit generated by federated operation both for TF and CS approaches (output of the maximization problems (13) and (14) respectively). Next, we present two profit-sharing rules, the Shapley-value driven and the activity-driven ones. In the former, we determine the profit that a CSP should earn based on its marginal contribution in the federation by making use of the Shapley value [16].

Shapley value has been widely used in coalitional game theory applications as a mechanism for sharing total utility in a fair manner (e.g., [18]), however as the number of players increases so does the computational complexity (the computation of Shapley-value is in general #P-complete [19]). Therefore, we propose an alternative heuristic technique of polynomial complexity. In particular, in the activity driven profit-sharing rule, the profit share that a CSP gets depends both on its standalone profit and on the level of involvement of the CSP in the task forwarding or capacity sharing actions either as supplier or receiver. We first present both profit sharing rules in the context of the TF federation mode and we then point out the differences in CS.

a) Shapley value driven profit-sharing [16]. A characteristic function $\upsilon(\cdot)$ measures the benefit of a coalition, also called the *worth of coalition*. In our approach, we take as characteristic function the total profit that is generated from the federation of a given set of CSPs under the optimal TF policy $\mathbf{A}^*$ derived by (13). In particular, the worth of coalition $\upsilon(\cdot)$ for the set of $\mathcal{N}$ CSPs is $\upsilon(\mathcal{N}, \mathbf{A}^*) = \sum_{i\in\mathcal{N}} P_i(\mathbf{A}^*)$. For a federation of N CSPs, the Shapley value of each CSP is obtained by calculating its average marginal contribution in all possible sub-federations $\mathcal{K} \subseteq \mathcal{N}$. Therefore, we need to compute the worth of coalition $\upsilon(\mathcal{K}, \mathbf{A}_K^*)$ for all possible subsets of CSPs $\mathcal{K}$. Note that $K = |\mathcal{K}|$ and $\mathbf{A}_K^*$ is the corresponding $K \times K$ dimensional matrix of forwarding policies. In order to derive the worth of coalition for all possible subsets $\mathcal{K}$, we solve the relevant optimization problems (13).

Assuming that $\mathcal{K} \subseteq \mathcal{N} \setminus \{i\}$, the *marginal contribution* of CSP i when it joins a sub-federation $\mathcal{K}$ is defined as

$$\mathcal{MC}_i(\mathcal{K}, \mathbf{A}^*_K, \upsilon) = \upsilon(\mathcal{K} \cup i, \mathbf{A}^*_{\mathcal{K}\cup i}) - \upsilon(\mathcal{K}, \mathbf{A}^*_{\mathcal{K}}) \quad (15)$$

Consequently, the profit share of a CSP i in the federation of N CSPs in TF is given by its *Shapley value* defined as

$$\varphi_i(\mathcal{N}, \mathbf{A}^*) = \sum_{\mathcal{K}\subseteq\mathcal{N}\setminus\{i\}} \frac{|\mathcal{K}|!\big(N - |\mathcal{K}| - 1\big)!}{N!} \mathcal{MC}_i(\mathcal{K}, \mathbf{A}^*_K, \upsilon) \quad (16)$$

where $\varphi_i(\mathcal{N}, \mathbf{A}^*)$ denotes the *estimated marginal contribution* of CSP i over all possible subsets of $\mathcal{K}$.

The only difference for applying this rule in CS is on the definition of characteristic function: $\tilde{\upsilon}(\mathcal{N}, \mathbf{B}^*) = \sum_{i\in\mathcal{N}} \tilde{P}_i(\mathbf{B}^*)$. The marginal contribution $\tilde{\mathcal{MC}}_i(\mathcal{K}, \mathbf{B}^*_K, \tilde{\upsilon})$ and Shapley value $\tilde{\varphi}_i(\mathcal{N}, \mathbf{B}^*)$ formulas are adjusted accordingly. As the number of CSPs increases so does the complexity of the Shapley value estimation [19], therefore we propose the activity driven profit-sharing policy as an alternative.

b) Activity driven profit-sharing. In this approach, a CSP i gets at least the profit it had in standalone operation, while the extra profit generated from the federated operation is proportionally shared among N CSPs based on the percentage of forwarded tasks that each of them forwarded or received. We define the *extra generated profit* $\Delta P(\mathbf{A}^*)$ by subtracting the aggregate profit of CSPs in the standalone operation from the total profit of federation

$$\Delta P(\mathbf{A}^*) = \sum_{i\in\mathcal{N}} P_i(\mathbf{A}^*) - \sum_{i\in\mathcal{N}} P_i(\mathbf{I}), \quad (17)$$

where $P_i(\mathbf{I})$ denotes the profit of CSP i in standalone operation. Consequently, the share of CPS i is determined by:

$$\xi_i(\mathbf{A}^*) = \frac{|\lambda'_i(\mathbf{a}'^*_i) - \lambda_i|}{\sum_{j\in\mathcal{N}} |\lambda'_j(\mathbf{a}'^*_j) - \lambda_j|} \Delta P(\mathbf{A}^*) + P_i(\mathbf{I}), \quad (18)$$

where $\frac{|\lambda'_i(\mathbf{a}'^*_i) - \lambda_i|}{\sum_{j\in\mathcal{N}} |\lambda'_j(\mathbf{a}'^*_j) - \lambda_j|}$ is the *proportionality* parameter which defines that a CSP who forwards or receives more tasks compared to another, will get a proportionally larger share of the extra generated profit.

In CS, the extra profit of a CSP is determined by the CSP's involvement in resource sharing action. Thus, the total extra profit generated $\Delta\tilde{P}(\mathbf{B}^*)$ by the CS federated operation is calculated by adjusting accordingly formula (17). Then, we can calculate the profit share $\tilde{\xi}_i(\mathbf{B}^*)$ of each CSP i by adjusting formula (18). Note that the proportionality parameter is now defined as $\frac{|C'_i(\mathbf{b}'^*_i) - C_i|}{\sum_{j\in\mathcal{N}} |C'_j(\mathbf{b}'^*_j) - C_j|}$. Note that in TF the contribution of the CSP that forwards a task is considered as equal to the contribution of the receiver, thus both will get the same amount of extra profit. The same applies in the CS approach for the CSP that either grants slices of infrastructure to others or is granted ones. This assumption makes sense because both the supplier and the receiver are important for the completion of a task forwarding (or resource sharing) action, and thus the generation of extra profit for the federation.

Remark II. The two profit-sharing rules differ on how the level of a CSP's contribution is perceived. In the activity driven sharing rule the extra profit is distributed only among the CSPs that are really involved in the task forwarding (or resource sharing) actions of optimal policy, either as supplier or receiver. On the other hand, Shapley value takes also into account the potential contribution of a CSP in all possible sub-federations. For more than two CSPs the two rules lead to different distribution of profits as we will see in section V. Finally, while the complexity of the Activity-driven profit sharing rule is polynomial, the Shapley-value driven rule belongs to the #P-complete class.

B. Reward-Driven Federation

Contrary to joint business mode, the reward-driven federation is suitable for selfishly acting CSPs. In this mode, each CSP determines its individual TF or CS policy aiming to maximize its profit, and thus a *non-cooperative game arises*. However, the game has been designed in such a way that motivates CSP to actively participate in the federation and serve the common good too. In particular, the payoff function of each CSP in this game incorporates the Shapley value as the reward mechanism that incentivizes them to contribute resources to the federation due to its fairness properties. Given this payoff function, each CSP adjusts its strategy to maximize its own payoff. Note that implementing Shapley value in a distributed way does not always achieve welfare maximization although it does in our case. It turns out that while CSPs act selfishly the outcome of this game also leads to social welfare maximization. Finally, the extent of the private information that the CSPs should share with others can be less in the present mode as we show below.

1) Task Forwarding-Based Reward-Driven Federation: Since the CSPs have conflicting objectives, it is not sufficient to define the individual profit of each CSP at the equilibrium as its payoff function, i.e. $P_i(\mathbf{A})$. Otherwise, selfish CSPs would be able to forward tasks without cost, thus leading the game to an equilibrium point where one or more CSPs would have lower profit compared to that in their standalone operation. As a result, CSPs that suffer losses would not have the incentive to participate. To meet this *participation constraint* for all CSPs (as also accomplished in the Joint business federation) and also to achieve a fair allocation of profits, it is announced to CSPs that their payoff from the federation in this game is determined by a fair contribution-based profit sharing rule, namely their Shapley value. Then, CSPs are left to play the game and choose their own TF policies.

Non-cooperative game. The set of players in this game is $\mathcal{N} = (1, 2, ..N)$. The individual TF strategy of a CSP i is defined by the entries of i-th row $\mathbf{a}_i$ of forwarding matrix $\mathbf{A}$, thus the set of CSPs' strategies is $\mathcal{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N)$. For CSP i we define by $\mathbf{a}_{-i}$ the strategies of all other CSPs except i. The payoff of each CSP in the game is determined by its Shapley value, thus the set of payoffs under a set of given strategies $\mathcal{A}$ is $\varphi = (\varphi_1(\mathcal{N}, \mathcal{A}), \varphi_2(\mathcal{N}, \mathcal{A}), \ldots, \varphi_N(\mathcal{N}, \mathcal{A}))$.

The game starts with each CSP operating in standalone mode, $\mathbf{A} = \mathbf{I}$. CSPs play in a round robin fashion in

each game iteration. In each step of the game, a CSP i determines its best response to the policies of all other CSPs. The best response of CSP i is a TF policy $\mathbf{a}_i$ that maximizes its payoff $\varphi_i(\mathcal{N}, \mathbf{a}_i, \mathbf{a}_{-i})$. Therefore, i determines its *best response* by solving the following optimization problem: $\arg\max_{\mathbf{a}_i} \varphi_i(\mathcal{N}, \mathbf{a}_i, \mathbf{a}_{-i})$. The constraints of this problem are the same as in the optimization problem of equation (13).

In order to calculate its Shapley value, a CSP has to compute its marginal contribution to all possible sub-federations $\mathcal{K} \subseteq \mathcal{N}$. For now, we assume that the necessary information for this computation is available and the game is played only for the full set $\mathcal{N}$ and not for subsets $\mathcal{K}$. Below, in Remark II, we discuss how the $\mathcal{MC}$ of CSP i in each sub-federation $\mathcal{K} \subseteq \mathcal{N} \setminus \{i\}$ can be obtained. The game continues until the system reaches a *Nash equilibrium* (NE) $\mathcal{A}^*$, where $\forall i \in \mathcal{N}$ and for every possible strategy $\mathbf{a}_i$, $\varphi_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*) \geq \varphi_i(\mathcal{N}, \mathbf{a}_i, \mathbf{a}_{-i}^*)$. In order to prove that the game converges to a NE and to characterize it, we employ the rationale followed in [18]:

Proposition I. *If, in each step of the game, a CSP i applies the individual forwarding strategy $\mathbf{a}_i^+$ that maximizes its payoff under Shapley value objective function, then the game converges to a state where the individually optimal strategies of all CSPs constitute the globally optimal solution $\mathcal{A}^*$, i.e. $\forall i \in \mathcal{N}$, $\mathbf{a}_i^+ = \mathbf{a}_i^*$. This set of strategies is a NE.*

PROOF: Given that under strategy $\mathbf{a}_i^+$ the $\varphi_i(\mathcal{N}, \mathbf{a}_i^+, \mathbf{a}_{-i})$ of CPS i is maximized. Due to strong monotonicity of Shapley value [16], the marginal contribution $\mathcal{MC}_i(\mathcal{N}, \mathbf{a}_i^+, \mathbf{a}_{-i}, \upsilon)$ of CSP i is also maximized by strategy $\mathbf{a}_i^+$. From (15), $\mathcal{MC}_i$ is maximized when the total profit of the subset that i joins is maximized. Consequently, in every step of the game, a CSP adapt its forwarding policy towards the maximization of the total federation profit. Hence, the game converges to the state $\mathcal{A}^*$ where $\forall i \in \mathcal{N}$, $\mathcal{MC}_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*, \upsilon)$ and thus $\varphi_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*)$ are maximized This state is a NE since none CSP can achieve higher payoff by changing its strategy. ■

2) Capacity Sharing-Based Reward-Driven Federation: Every CSP that participates in a CS reward-driven federation knows that its payoff is determined by its Shapley value, thus each of them determines its CS policy aiming to maximize it.

Non-cooperative game. The only significant differences compared to TF approach is on the strategies of the players and their payoff functions. In particular, the strategy of CSP i is its CS policy defined by the i-th row of global resource sharing matrix $\mathbf{B}$, while its payoff is given by $\tilde{\varphi}_i(\mathcal{N}, \mathbf{b}_i, \mathbf{b}_{-i})$. Therefore the best response of CSP i is determined by the following maximization problem: $\arg\max_{\mathbf{b}_i} \tilde{\varphi}_i(\mathcal{N}, \mathbf{b}_i, \mathbf{b}_{-i})$. Following a similar approach to the case of TF Reward-driven federation, we can prove that the game converges to a unique NE $\mathcal{B}^*$ which is also globally optimal.

Remark III. In order to determine its best response in the previously presented games, each CSP should calculate its Shapley value based on its marginal contribution in all sub-federations $\mathcal{K} \subseteq \mathcal{N}$. There are two alternatives to obtain this information: (i) All CSPs play recursive non-cooperative games as above for all possible sub-federations. They start playing these games from the smallest to largest subset, and the output of each game is used as input to the larger ones. (ii) Same as in subsection IV-A, each CSP solves the relevant global optimization problem for all subsets $\mathcal{K}$ and uses the results as input on determination of its best response in a unique game for the full set of $\mathbf{N}$ CSPs. Note that in the first approach, we have multiple games and thus higher complexity, but CSPs should only reveal private information that is limited to their profit in the standalone operation. The second approach has lower complexity because we only have one game. However, this approach has the *drawback* that each CSP should reveal more *private information* that includes its computational capacity and average task request load, as done in joint business federation, in order for each of the other CSPs to solve the optimization problem giving its best response.

V. Numerical Evaluation

A. Simulation Setup

We simulate an environment of three CSPs that can operate under all presented federation approaches, modes and policies.

CSPs dimensioning: CSPs are symmetric with respect to their computational capacities, which equals $C = 2$ Tera·operations per second. This capacity corresponds to typical 100 servers and can support tasks of an average arrival rate of 10 tasks/sec. However, this capacity is not fully utilized since the highest infrastructure utilization is determined by the worst acceptable QoS that each CSP guarantees to its customers. To this end, we assume that all CSPs have the same worst acceptable QoS $d^{max} = 1$ (sec/task).

Task request arrivals: We assume that CSPs 2 and 3 have a fixed rate of incoming tasks λ_2 and λ_3, while λ_1 takes values in the range $[1, \lambda_1^{max}]$, where λ_1^{max} is extracted from d_{max}. Since the CSPs are identical $\lambda_1^{max} = \lambda_2^{max} = \lambda_3^{max}$. We run experiments of this type for different fixed values of λ_2 and λ_3, both in the interval 1 to λ_1^{max}.

Task size: We assume that each task that arrive in a CSP requires an average of $L = 200$ Giga·ops.

Power consumption: For the power consumption, we take the idle and full utilization powers as $W_0 = 60$ KWatt and $W_1 = 400$ KWatt. We assume that all CSPs pay the same price to their electricity provider, namely $z = 2.7{\cdot}10^{-5}$ \$/KWatt·sec. This value corresponds to 0.1 \$/KWh, which is a typical price.

Pricing: The three CSPs charge their customers according to the same pricing function, with same maximum price q \$/task. In our experiments, we select the value of q by taking as input the electricity price z. In particular, given the price z, we find the value of q for which the profit of CSP becomes zero when the utilization factor is 0.99. This guarantees that the CSPs in standalone operation will have some profits for any value of utilization up to 99%. Thus, the price per task in our setup is $q = 0.11$ \$/task.

Network delay: The additional communication delay D for the task requests transfered over the Internet is taken to be $D = 10$ msec. However, we also explore how different values of it affect the impact of the TF and CS federation modes.

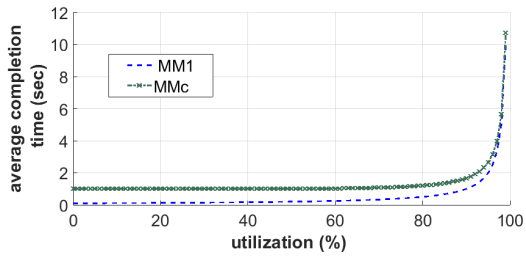


Fig. 3. Average task completion time for a CSP under different utilizations levels for both $M/M/1$ and $M/M/c$ modeling approaches.

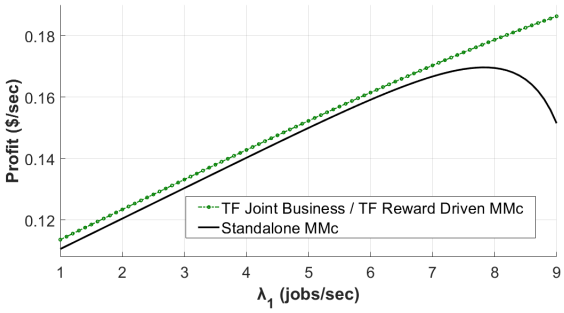


Fig. 4. Total profit of CSPs under $M/M/c$ model and all modes of TF federation, for $\lambda_2 = 7$, $\lambda_3 = 4$ and $\lambda_1 \in [1, 9]$. Note that $d^{max} = 1$ (sec/task), which implies that the maximum arrival rate that a CSP can have equals 9.

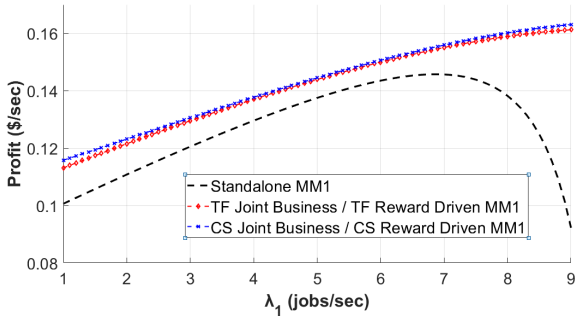


Fig. 5. Total profit of CSPs under $M/M/1$ model and all operation modes, for $\lambda_2 = 7$, $\lambda_3 = 4$ and $\lambda_1 \in [1, 9]$. Note that $d^{max} = 1$ (sec/task), which implies that the maximum arrival rate that a CSP can have equals 9.

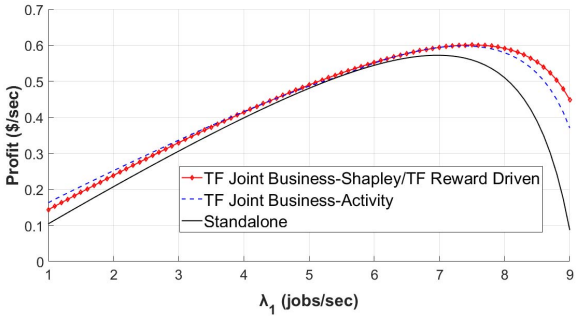


Fig. 6. Individual profit of CSP 1 under different TF federation operation modes, for $\lambda_2 = 7$, $\lambda_3 = 4$ and $\lambda_1 \in [1, 9]$.

B. Numerical Results

1) Impact of M/M/1 Abstraction: In this paragraph, we justify through numerical results why abstracting a CSP to an $M/M/1$ queueing system is equally reasonable to an $M/M/c$ modeling approach. As we already mentioned in section II-A, we argue that both $M/M/1$ and $M/M/c$ queueing systems are applicable in our federation model since the *average task completion time* in both modeling approaches is a convex and increasing function of the task arrival rate λ, as illustrated in Fig. 3. In order to further justify that our claim, we have implemented and performed multiple experiments for TF federation mode under both $M/M/1$ and $M/M/c$. As depicted in Fig. 4 and Fig. 5, our federation models and policies achieve qualitatively the same outcome in terms of total TF federation profit. This also applies for CS model considering results related to the CSPs' individual profits, TF/CS policies and customers QoS. All these aspects are discussed to the reminder of this section under $M/M/1$ model.

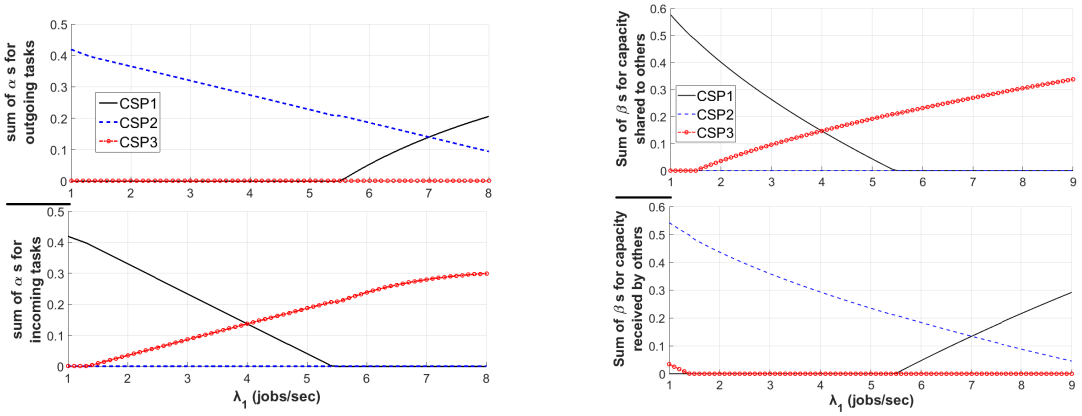
2) Total Profit: Figure 5 shows the total profit under all operation modes, for fixed value of $\lambda_2 = 7$, $\lambda_3 = 4$ and for $\lambda_1 \in [1, 9]$. Note that the arrival rate threshold of each CSP is $\lambda_1^{max} = 9$. The results reveal that both the TF and CS approaches can achieve significantly higher total profit compared to the aggregate profit of CSPs in standalone operation. The only case where the total federation profit can be equal to the aggregate standalone profit is when all CSPs have the same standalone utilization level because then, e.g in TF, the optimal solution is $\mathbf{A}^* = \mathbf{I}$. Furthermore, the total profit of joint business and reward-driven modes appear to coincide for both the TF and CS approaches and under all possible values of λ's. This happens because of the use of Shapley value as a CSP's payoff in the reward driven mode, since Shapley value leads each CSP to act for the benefit of the federation as a whole.

Comparing the two federation approaches, we notice that the CS can achieve slightly higher total profit for all values of λ's. This applies because in CS federation we assume pooling of both owned and shared resources. As we mentioned in Remark I, this means that a CSP can scale up its infrastructure and offer higher QoS compared to the standalone or TF federation, thus generating higher revenues. Later in this section we will show that this becomes more prominent if the selected pricing function is "unfair" as defined in Discussion I. Finally, the more diverse the CSPs' standalone infrastructure utilization values, the more pronounced the benefit of both TF and CS federation modes. This is apparent in Fig. 5 where the extra generated profit compared to standalone is higher for $(\lambda_1, \lambda_2, \lambda_3) = (9, 7, 4)$ than for $(\lambda_1, \lambda_2, \lambda_3) = (8, 7, 4)$.

3) Individual CSP Profit: The individual profit of each CSP in all federation approaches and modes is by design higher than or at least equal to its profit in standalone operation. Fig. 6 shows the individual profit of CSP 1 under all possible operation modes in both TF approach. The results show that the individual profits that a CSP earns under both TF approach are always higher than its profit under the standalone operation. The results are quite similar for CS approach.

In addition, both for CS and TF approaches, the profit share of each CSP is the same when the federation operates under the reward-driven or under joint business mode with Shapley value as profit sharing rule. This happens because of the use of Shapley value as payoff function of each CSP in the game.

Comparing the two profit sharing policies of joint business mode, we highlight the following: (i) The activity driven profit



(a) Optimal TF policies (b) Optimal CS policies

Fig. 7. CSPs' optimal TF and CS policies, for $\lambda_2 = 7$, $\lambda_3 = 4$ and $\lambda_1 \in [1, 9]$. The top part of Fig. 7a shows the sum of the probabilities that determine the portion of tasks each CSP forwards under TF mode, while the lower part shows the sum of the probabilities are responsible for the portion received tasks. Fig. 7b shows the sum of probabilities for the shared (top) and granted (bottom) capacity for each CSP under CS mode.

sharing policy favors the less utilized CSPs more than the Shapley value driven one. This is seen in Fig. 6 for low values of λ_1, where the profit CSP 1 is higher under the activity driven policy. (ii) The activity driven policy may give to a CSP just the profit it had in the standalone operation. This happens when a CSP does not participate in the forwarding/sharing actions either as supplier or receiver.

4) TF/CS Policies: The results again show that the optimal TF policy is the same for both joint business and reward driven federation modes. Interestingly, the same applies to the optimal CS policy. Fig. 7a is related to the TF approach and the optimal TF policy. We observe that in the optimal TF policy either the sum of the α probabilities that determine the portion of tasks that each CSP forwards or the sum of the probabilities are expressing the portion of forwarded tasks arrive at each CSP queue will be equal to zero (can be both zero). This means that every CSP either forwards or receives tasks but never does both. The above observation also applies to the CS approach (Fig. 7b), where a CSP that shares part of its infrastructure does not utilize capacity of others. Furthermore, we have observed that both in TF and CS approaches all CSPs that forward tasks/grant resources have a utilization level that is higher than the federation's average, while CSPs that receive tasks/share resources have a lower utilization than the federation's average.

5) QoS Level: For comparison purposes, we consider the TF and CS QoS-optimal federation modes, which employ those CS/TF policies that optimize the average QoS of the federation. Fig. 8 shows the average QoS of whole federation customers under all modes. The results reveal that under both TF and CS approaches all modes achieve the same performance in terms of QoS and outperform the QoS of standalone operation. Interestingly, the performance of all TF and CS modes are close to the respective QoS-optimal. Regarding the QoS of individual CSP's customers, it is close to total average because both TF and CS federation achieves load balancing.

6) Impact of Communication Delay: We now investigate how the CSPs communication delay (D) affects the performance of both TF and CS federation modes. Fig. 9 shows that as D increases, the total profits of both TF and CS federation

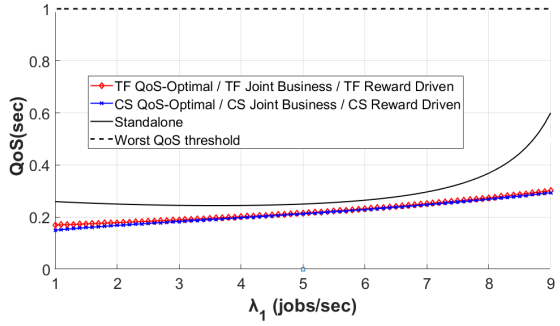


Fig. 8. Customers' QoS under different operation modes, for $\lambda_2 = 7$, $\lambda_3 = 4$ and $\lambda_1 \in [1, 9]$. Worst QoS threshold is set at $d_i^{max} = 1$ (sec/task).

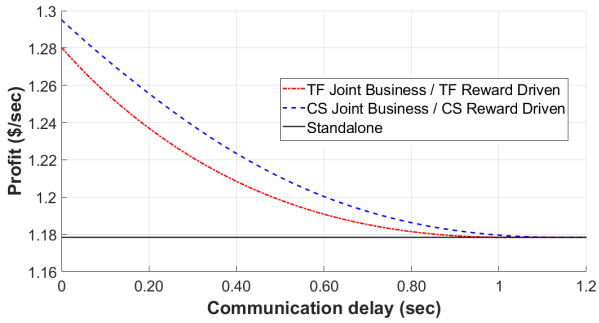


Fig. 9. Total profit of CSPs under all operation modes, for $\lambda_1 = 2$, $\lambda_2 = 7$, $\lambda_3 = 4$, $d^{max} = 0.5$, and for $D \in [5, 1200]$ msec.

modes diminish. Then D exceeds a certain value, the profits of participating CSPs becomes equal to the one of standalone operation. This happens because as D increases the CSPs follow a more conservative TF (or CS) policies, thus under very high values of D the optimal choice is to not forward tasks (or share capacity) and fall back to standalone operation. Consequently, network delay turns out to be an important parameter for the effectiveness of both TF and CS modes.

7) Asymmetric CSPs: We also evaluate our models for asymmetric CSPs with respect to their infrastructure dimensioning. We run the same type of experiments considering different sizes for the infrastructure of CSP 2. In particular, we consider the following three setups of CSPs' dimensioning $(C_1, C_2, C_3) = \{(2, 1, 2), (2, 2, 2), (2, 4, 2)\}$, where the values

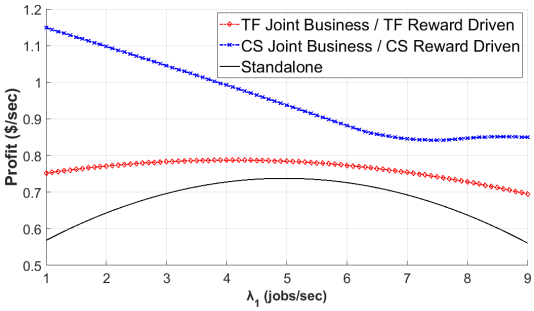


Fig. 10. Total CSPs profit all operation modes and "unfair" pricing.

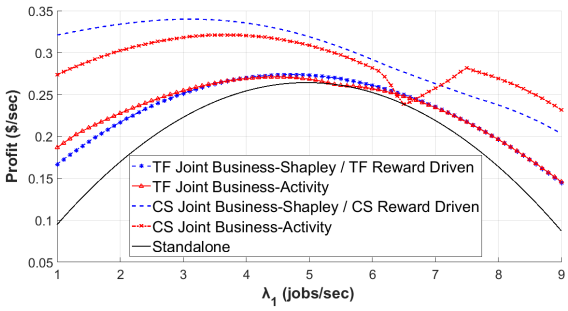


Fig. 11. Profit of CSP1 under all operation modes and "unfair" pricing.

refer to Tera·operations per second. Since we assume that the power consumption of infrastructure is related to its computational capacity, the CSPs are also asymmetric with respect to their power consumption. We consider the three following pair of values for the idle and dynamic power consumption of the CSP $\{(30, 200), (60, 400), (120, 800)\}$, for 1, 2 and 4 Tera·operations per second dimensioning respectively. The arrival rates we consider in each setup are selected in way that maintains the utilization level of each CSP regardless of its dimensioning, e.g. if for $C_1 = 2$ the arrival rate was set $\lambda_2 = 6$, then for $C_1 = 4$ we should set $\lambda_2 = 12$.

The results reveal that our models works perfectly also for asymmetric CSPs. The impact of infrastructure asymmetry is summarized in the following points: (i) In terms of total federation profit, the benefit of TF mode remains the same regardless of the infrastructure dimensioning when the workload increases respectively. (ii) In terms of individual CSP profits, the results show that large CSPs that have higher standalone utilization compared to other federation members can have more benefit than in the opposite case but the federation remain beneficial for all participants.

8) Impact of "Unfair" Pricing Functions: In this paragraph we investigate the impact of "unfair' pricing functions (as defined in Remark I) on the behavior of CSP under CS federation. A pricing function that satisfies this criterion, i.e. renders the CSP's revenue R_i a convex functions to service rate μ_i, is $p_i(\lambda_i) = \frac{d_i^{min}}{d_i(\lambda_i)} q_i$. The results reveal that by adopting such a function, the CS federation can be lead to completely different results both in terms of CSPs' profits and customers' QoS. As shown in Fig. 10, the total profit of CSPs under CS federation becomes significantly higher. This happens because now CS federation abuses the "unfair" pricing function. In particular, the optimal CS policy is to increase the infrastructure of the CSP with the highest workload as much as possible, leading to a really high QoS and thus price for the customers of this CSP. On the other hand, the rest CSPs operate at the worst acceptable QoS and their customers pay the minimum price.

The impact of "unfair" pricing on the behavior of CS federation is summarized in the following points: (i) Under CS joint business mode, the activity-driven profit sharing policy now favors the CSPs with higher utilization (see high values of λ_1 in Fig. 11). (ii) In CS federation, the QoS and profit optimization objectives are not aligned. Now, CS federation can achieve higher profits than TF at the expense of lower average QoS which in some case becomes worse than that of standalone operation. In fact, the average QoS in now

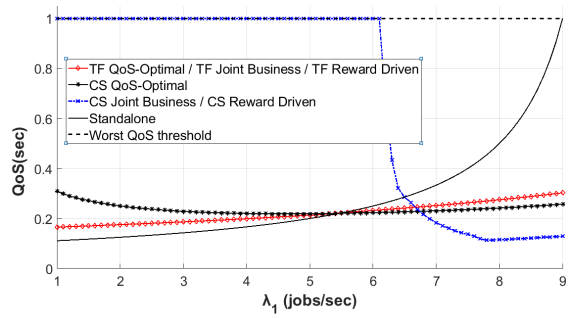


Fig. 12. QoS of CSP1 under all operation modes and "unfair" pricing.

driven by the worst acceptable QoS thresholds of CSPs. In the optimal policy, two of the CSPs operate at the worst acceptable QoS, while the third one operates at very high QoS. Fig. 12 shows that CSP 1 switches from the worst QoS level to a very high one as its standalone workload increases. (iii) As the QoS thresholds become tighter, the average QoS of the CS approach improves but the generated profits decrease. The existence of worst QoS threshold is vital for the CS approach, otherwise the optimal CS policy would lead to the maximum possible profit but also to unacceptable QoS. Next, we discuss how the deficiency occurs by the "unfair" can be mitigated by taking into account customers' *willingness to pay*.

9) Restriction on Best Possible QoS and Customers' Willingness to Pay: In order to avoid an extremely high QoS and an undesirably high price for the customers, each CSP could define a best QoS threshold based on its customers' *willingness to pay*. That is when a customer sends a task request to a CSP, he should also define the maximum price p^{max} he is willing to pay for this task. For simplicity, let us assume that this is the same for all customers. The CSPs should take as input this p^{max} and estimate the T^{min} above which is still beneficial to work. Operating at better QoS level will not bring additional profit because the customers are not willing to pay for it. We performed multiple experiments by adjusting slightly our models to introduce willingness pay dimension. The results show that the TF approach is not affected by this new feature of the model. On the other hand, the profitability and QoS of the CS approach now depends on the new best QoS threshold that is determined by customers' willingness to pay, i.e. p^{max}. Especially, the total profit of CS is slightly lower than before but the QoS is slightingly better. This applies because the new upper threshold on QoS places a restriction on how much QoS can be "traded" for extra profit.

VI. Related Work

Architectural approaches of cloud federation. The authors in [10] present the challenges of a utility-oriented cloud federation and propose three basic entities for a market-based cloud federation architecture; namely, the cloud exchange as the entity that creates the market, a cloud coordinator per CSP as seller, and a cloud broker per client as buyer. The Reservoir model, a modular cloud architecture, is proposed in [20]. In Reservoir, multiple CSPs collaborate in order to create a virtual pool of resources that seems infinite. The authors in [21] present the concept of cloud federation as service aggregation and they present two modes of such a federation, the redundancy and migration federations. In redundancy federation, multiple CSPs come together and jointly offer a service to achieve improved quality for a client, while in migration federation a client is moved from an old service to new one offered by another CSP due to improved quality (this condition for migration constitutes the main difference to our TF approach). Finally, the authors in [22] envision the federation of CSPs as a vertical stack that fits on the layered model of cloud computing (i.e. SaaS, PaaS and IaaS). A service request may arrive in any layer of a CSP and can be served either by local resources using delegation to a lower layer or by another federated CSP using delegation to a matching layer.

Cooperative inter-cloud resource allocation. The authors in [23] and [24] propose cooperative price-based resource allocation mechanisms in dynamic cloud federation platforms, aiming to maximize the total utility of a federation. In [7], [25] and [8], coalitional game theory is applied as a mechanism for the dynamic formation of CSPs' federations. Both these papers have proposed algorithms that determine the optimal coalitions for a set of CSPs, given their client-generated workloads. A work that is also based on the coalitional game theory was presented in [26], however this works is focusing in data intense federations and incorporates trust models for the coalition formation as well penalties for SLA violation. In [9] the inter-CSP VM (virtual machine) migration is presented as an alternative to resource over-provisioning. The authors propose a global scheduler that decides whether a VM should migrate or shut down, thus aiming to CSPs utility maximization.

Resource allocation among selfish CSPs. In [11], the federation among geo-distributed CSPs is investigated. The authors design algorithms, based on double-auctions, for inter-cloud VM trading in federations of selfish CSPs. A Stackelberg game is presented in [12]; the game is between the Application Service Providers (followers) that aim at optimizing their offered QoS and the CSPs (leaders) that set prices of resources to maximize their own benefit.The authors in [27] model each CSP as a set of heterogeneous servers, each of them modeled as an $M/M/1$ queue. Then, they formulate the problem of resource allocation in a multi-CSP environment as a game among selfish CSPs, where each CSP aims to maximize its individual utility taking into account the customer SLAs. The author in [13] investigates the interactions among CSPs as a repeated game among selfish players that aim at maximizing their profit by selling their unused resources in a spot market. The model incorporates information for both historical and expected future revenue as part of the resource trading decisions, in order to simultaneously maximize the CSP revenue and avoid future workload fluctuations.

Some of the works above provide an overview of the architectural elements of a federated system, while others consider the problem of resource allocation in inter-cloud environments of either cooperative or selfish CSPs. Contrary to most existing works, we provide policies both for cooperative and the non-cooperative federated environments. Most of the existing works do not take into account the QoS offered to CSPs' customers as the optimization objective. In our TF approach, the federation policies are optimal with respect to total profit; but they are also nearly optimal with respect to the QoS offered to customers since profit and QoS optimization are aligned objectives. In the CS approach, the worst QoS threshold gives federated CSPs the ability to generate higher profits by reducing the average QoS as close as possible to the customers' SLAs. Finally, our work abides to the general trend observed in the area of cloud computing such "serverless computing" which appears to be the service model that will dominate the cloud market in the forthcoming years.

VII. Conclusions

In this paper, we built a fundamental theory of sharing economy of computational capacity resources of CSPs. In particular, we introduced a queueing theory-driven model for a CSP and we formulated its revenue and cost functions. We defined two alternative approaches (TF and CS) for the formation of cloud federations, and we introduced policies and rules for two different federation modes. In the joint business mode, the CSPs cooperate and jointly decide their TF/CS policies aiming to maximize the total profit of federation. In the reward driven mode, the CSPs participate in a non-cooperative game where each of them determines its own TF/CS policy aiming to maximize his own profit. However, the game is designed in a way that motivates selfish CSPs to contribute to the federation.

The numerical results showed that our models can considerably increase the profit of the participating CSPs and federation as a whole. An important outcome of our work is that the reward-driven federation of selfish CSPs converges to a unique equilibrium where the CSPs' profits are the same as in the cooperative joint business federation. This implies that the optimal solution can be achieved both for cooperative and selfish CSP groups. We also showed that the CS federation approach can generate significantly more profit compared to the TF under certain pricing function. However, in that case,the QoS and profit optimization objectives under CS approach are not aligned. This should be taken into account by CSPs aiming to create a federation. Base on the CSPs interests (e.g. high profit, high QoS, etc.), the pricing function that they follow can render both TF and CS federation approaches applicable, or only one of them.

An interesting future direction is to generalize these models in service provisioning environments, where the service provisioned requires computational capacity, storage and bandwidth resources which may be owned by different parties.

ACKNOWLEDGMENT

The authors wish to express their thanks to Prof. R. T. B. Ma (National University of Singapore) for useful discussions on the subject of the paper.

REFERENCES

[1] *Federation is the Future of the Cloud*. Accessed: Jan. 24, 2017. [Online]. Available: http://www.datacenterknowledge.com/archives/2012/09/17/federation-is-the-future-of-the-cloud/
[2] *OnApp Federation*. Accessed: Feb. 17, 2016. [Online]. Available: http://onapp.com/federation
[3] *Arjuna Agility*. Accessed: Feb. 16, 2016. [Online]. Available: http://www.arjuna.com/federation
[4] *EGI Federated Cloud*. Accessed: Feb. 15, 2016. [Online]. Available: https://www.egi.eu/infrastructure/cloud
[5] *BonFIRE Project*. Accessed: Feb. 17, 2016. [Online]. Available: http://www.bonfire-project.eu
[6] CERN. *OpenLab Project*. Accessed: Mar. 1, 2016. [Online]. Available: http://openlab.web.cern.ch
[7] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 14–27, Jan./Mar. 2015.
[8] M. Guazzone, C. Anglano, and M. Sereno, "A game-theoretic approach to coalition formation in green cloud federations," in *Proc. 14th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2014, pp. 618–625.
[9] I. Goiri, J. Guitart, and J. Torres, "Characterizing cloud federation for enhancing providers' profit," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010, pp. 123–130.
[10] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.*, 2010, pp. 13–31.
[11] H. Li, C. Wu, Z. Li, and F. C. M. Lau, "Profit-maximizing virtual machine trading in a federation of selfish clouds," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 25–29.
[12] H. Roh, C. Jung, W. Lee, and D.-Z. Du, "Resource pricing game in geo-distributed clouds," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1519–1527.
[13] N. Samaan, "A novel economic sharing model in a federation of selfish cloud providers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 12–21, Jan. 2014.
[14] G. Darzanos, I. Koutsopoulos, and G. D. Stamoulis, "A model for evaluating the economics of cloud federation," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 291–296.
[15] G. Darzanos, I. Koutsopoulos, and G. D. Stamoulis, "Economics models and policies for cloud federations," in *Proc. IFIP Netw. Conf. (IFIP Netw.) Workshops*, May 2016, pp. 485–493.
[16] L. S. Shapley, "A value for n-person games," in *Contributions to the Theory of Games* (Annals of Mathematics Studies 28), vol. 2. Princeton, NJ, USA: Princeton Univ. Press, 1953, pp. 307–317.
[17] M. Steinder, I. Whalley, J. Hanson, and J. Kephart, "Coordinated management of power usage and runtime performance," in *Proc. NOMS*, Apr. 2008, pp. 387–394.
[18] R. T. B. Ma, D. M. Chiu, J. Lui, V. Misra, and D. Rubenstein, "Internet economics: The use of Shapley value for ISP settlement," *IEEE/ACM Trans. Netw.*, vol. 18, no. 3, pp. 775–787, Jun. 2010.
[19] U. Faigle and W. Kern, "The shapley value for cooperative games under precedence constraints," *Int. J. Game Theory*, vol. 21, no. 3, pp. 249–266, Sep. 1992.
[20] B. Rochwerger *et al.*, "Reservoir—When one cloud is not enough," *Computer*, vol. 44, no. 3, pp. 44–51, Mar. 2011.
[21] T. Kurze *et al.*, "Cloud federation," in *Proc. Cloud Comput.*, vol. 2011, pp. 32–38, Sep. 2011.
[22] D. Villegas *et al.*, "Cloud federation in a layered service model," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1330–1344, 2012.
[23] M. M. Hassan, B. Song, and E.-N. Huh, "Distributed resource allocation games in horizontal dynamic cloud federation platform," in *Proc. IEEE Int. Conf. High Perform. Comput. Commun.*, Sep. 2011, pp. 822–827.
[24] S. Rebai, M. Hadji, and D. Zeghlache, "Improving profit through cloud federation," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2015, pp. 732–739.
[25] D. Niyato, A. V. Vasilakos, and Z. Kun, "Resource and revenue sharing with coalition formation of cloud providers: Game theoretic approach," in *Proc. 11th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2011, pp. 215–224.
[26] M. M. Hassan *et al.*, "QoS and trust-aware coalition formation game in data-intensive cloud federations," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 10, pp. 2889–2905, Jul. 2016.
[27] Y. Wang, X. Lin, and M. Pedram, "A game theoretic framework of SLA-based resource allocation for competitive cloud service providers," in *Proc. 6th Annu. IEEE Green Technol. Conf.*, Apr. 2014, pp. 37–43.

George Darzanos received the B.Sc. in informatics and telecommunications from the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece, in 2011, and the M.Sc. in computer science from the Department of Informatics, Athens University of Economics and Business (AUEB), in July 2013, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. He is also a member of the Services Technologies and Economics Group (STEcon Group). His research interests include the design of incentive mechanisms for economic management of traffic generated by cloud-based applications. Finally, part of his research covers the design of collaboration and coordination schemes for services composition in the 5G multi-actor ecosystem.

Iordanis Koutsopoulos (S'99–M'03–SM'13) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Greece, in 1997, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park (UMCP), MD, USA, in 1999 and 2002, respectively. He has served as an Assistant Professor with AUEB from 2013 to 2016 and an Assistant Professor from 2010 to 2013 and a Lecturer from 2005 to 2010 with the Department of Electrical and Computer Engineering, University of Thessaly. He has been an Associate Professor with the Department of Informatics, Athens University of Economics and Business (AUEB), Athens, Greece, since 2016. His research interests are in the general area of control and optimization and on applications of machine learning, with application areas, such as mobile crowdsensing, wireless networks, social networks, recommender systems, smart energy grid, and cloud computing systems. He received the Single-Investigator European Research Council (ERC) competition Runner-Up Award for the project RECITAL: Resource Management for Self-Coordinated Autonomic Wireless Networks from 2012 to 2015 and the three best paper awards for research on online advertising, network economics, and network experimentation.

George D. Stamoulis received the Diploma degree (Hons.) in electrical engineering from the National Technical University of Athens, Greece, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1988 and 1991, respectively. He is currently a Professor with the Department of Informatics, Athens University of Economics and Business (AUEB), the Head of the Technology Economics [Service, Technologies, and Economics (STEcon)] Group, and the Dean of the School of Information Sciences and Technology, AUEB. He has coordinated the participation of the NES Group, AUEB, in several successful H2020 and FP7 projects. He has published over 100 articles in scientific journals, including the IEEE TRANSACTIONS ON CONTROL OF NETWORKED SYSTEMS, the IEEE/ACM TRANSACTIONS ON NETWORKS, *Computer Communications*, the IEEE TRANSACTIONS ON COMMUNICATIONS, *Computer Networks*, and the *Journal of the ACM*, and in scientific conferences, including INFOCOM, ITC, ACM SIGMETRICS, IFIP Networking, and e-Energy. He has also collaborated with Greek Regulatory Authorities for Communications and Power on auction design as well as with Greek telecom companies on service pricing and data analysis. His research interests are in economic and business models for networks, clouds and smart grids, Internet traffic management with emphasis on the use of economic mechanisms, auction mechanisms for bandwidth and scarce goods, demand response in energy consumption, telecommunications and power regulation, and reputation mechanisms for electronic environments.