Support Service for Reciprocal Computational Resource Sharing in Wireless Community Networks

Ümit C. Büyükşahin, Amin M. Khan, Felix Freitag

Department of Computer Architecture
Universitat Politècnica de Catalunya
Barcelona, Spain
Email: {ubuyuksa, mkhan, felix}@ac.upc.edu

***Abstract*—In community networks, individuals and local organizations from a geographic area team up to create and run a community-owned IP network to satisfy the community's demand for ICT, such as facilitating Internet access and providing services of local interest. Most current community networks use wireless links for the node interconnection, applying off-the-shelf wireless equipment. While IP connectivity over the shared network infrastructure is successfully achieved, the deployment of applications in community networks is surprisingly low. To address the solution of this problem, we propose in this paper a service to incentivize the contribution of computing and storage as cloud resources to community networks, in order to stimulate the deployment of services and applications. Our final goal is the vision that in the long term, the users of community networks will not need to consume applications from the Internet, but find them within the wireless community network.**

***Index Terms*—wireless mesh networks; community networks; cloud computing; incentive mechanisms**

I. INTRODUCTION

Wireless community networks are an emergent model of infrastructure that aims to satisfy a community's demand for Internet access and ICT services. Most community networks originated in rural areas which commercial telecom operators left behind when deploying the broadband access infrastructure for the urban areas. Different stakeholders of such a geographic area teamed up to invest, create and run a community network as an open telecommunication infrastructure based on self-service and self-management by the users [1].

Current community networks use mainly wireless technology to interconnect nodes. With the commoditization of optical fiber, some community networks however have also started providing broadband services combining both technologies (e.g. guifi.net with fiber to the farm, FTTF [2]).

Community networks share, to a greater or lesser extent, the following common characteristics:

- they apply network neutrality such that the bandwidth capacity is limited only by the physical constraints of the deployed technologies.
- are public utilities available for use on equal terms by any party (private, public, commercial) connected to it within the community it serves.
- provide infrastructure which on the macro-level is community-owned, while on the micro-level of equipment is owned by the individual participants that contributed it.

Community networks are a successful case of resource sharing among a collective. The resources shared are networking hardware but also each community network participant's time he/she donates, in different extent, for maintaining the network. While the community network infrastructure is the sum of the individual contributions of wireless equipment, the network operation is achieved by the contribution of time and knowledge of the participants, even under the decentralized management of the equipment, since the node owner ultimately has the full access and control of his/her network device.

Resource sharing in community networks from the equipment perspective refers in practice to the sharing of the nodes' bandwidth. This sharing enables that traffic from other nodes is routed over the nodes of different node owners. This is done in a reciprocal manner which allows community networks to successfully operate as IP networks. Computing and storage resource sharing, such as is now common practice in today's Internet through cloud computing, hardly exists in community networks. So any service offered in community networks runs on machines exclusively dedicated to a single member.

In Figure 1, some node types of a community network are depicted. The picture shows typical community nodes with a router and some server (or client machine) attached to it. A community network distinguishes between super nodes and client nodes. Super nodes have at least two wireless links, each to other super nodes. Some super nodes are placed strategically in some geographic area to improve the community network's backbone and thus consist only of the wireless router. Other super nodes are installed in the community network participant's premises. In that latter case, such as shown in some nodes in Figure 1, servers behind the router are connected to offer services and applications to the community network. Client nodes only connect to a super node, but do not route any traffic. In Figure 1, some client nodes are shown which are connected to the access point (AP) of a super node. Topological analysis of the Guifi.net community network [3] indicates that from approximately 17000 analysed nodes of Guifi.net, 7% are super nodes while the others are client nodes.

From the node types shown in Figure 1 it can be seen

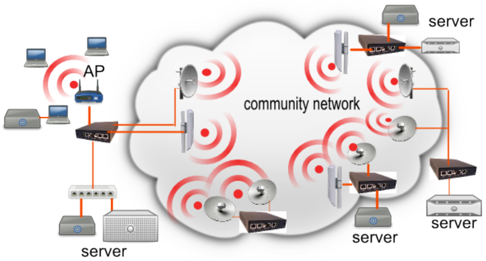

Figure 1. Nodes in a Community Network

that principally the hardware for computation and storage is already available in community networks, consisting of some servers attached to the wireless routers. No cloud services, however, are yet deployed in community networks to use this hardware as a cloud, leaving the community network services significantly behind the current standard of the Internet. Our vision is that some community wireless routers will have cloud resources attached, building the infrastructure for a community cloud formed by several cloud resources attached to community nodes. We note that client nodes could principally also contribute cloud resources. We therefore centre the contribution of this paper towards how to incentivize to bring together these computation and storage hardware already attached to the wireless routers of the community network into a community cloud.

We propose a support service to incentivize reciprocal computational resource sharing, which we envision to be part of a future community cloud service infrastructure. This component should incentivize and stimulate the sharing of resources in community networks, extending what is already done in wireless community networks by sharing of bandwidth, to computational and storage resources. The main functionality of this support service is achieving a regulation of the contribution and consumption of resources by users, such that a participant would benefit from improved service experience if he/she makes a larger contributions of resources.

In the following sections we present our proposal of the main components that the architecture for a community cloud should have. A key component of this architecture is the support service for achieving reciprocal resource sharing, which we address in this paper. In section II we describe how this architecture would be applicable to the topology of current community network deployments in which the interconnection and traffic routing is done by super nodes. In section III we detail the algorithm that we propose for the reciprocal resource sharing service to regulate resource consumption as a function of contribution. Considering different resource sharing scenarios, we define a model that represents such resource sharing systems. With simulations we explore the parameter space of our algorithm and show the performance effects of different parameter values in section IV. In section V we discuss the related work, and in section VI we conclude our findings and discuss about future work.

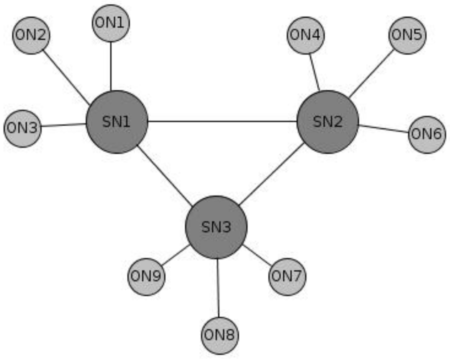

Figure 2. Overlay Network of Super Nodes and Ordinary Nodes in a Community Cloud

II. DISTRIBUTED ARCHITECTURE USING SUPER NODES

A. *Context of the community network topology*

The community mesh network generally includes different node types and each type plays a different role in the network. For example, Guifi.net [4], which is considered the largest meshed community network worldwide, includes two main types of nodes, according to [3]: *terminal nodes* which represent the end user nodes, and *hubs* which serve traffic to end users. In this network, each terminal node has a unique connection to a hub that routes traffic, and hubs can have many connected terminal nodes [3].

The architecture of a community cloud has to consider the topology of the community network which the cloud will be deployed on. Considering the typical community nodes explained above and the analysis of the community network topology [3], a hierarchical architecture [5] for community clouds is suggested. In this architecture, each super node is responsible for the management of a set of attached nodes. From the perspective of the attached nodes, these super nodes act as a centralized unit to manage the cloud services. These super nodes connect physically between other super nodes and logically in an overlay network to other cloud managing nodes.

This hierarchical architecture can be classified into the two main classes of fully decentralized and centralized systems [5]. If the design of the architecture is done towards a centralized systems, advantages include efficient search and control, while if it is decentralized, load-balancing, robustness and failure tolerance would be the benefits. There are several large-scale distributed applications that using a hierarchical design achieved great success, such as Kazaa and Skype.

B. *Architecture and design*

Figure 2 depicts the overlay network that results from the hierarchical architecture of the community cloud, having ordinary nodes (ON) and super nodes (SN). ONs behave both as provider and requester in the cloud system. That means, at different times they can both request a resource or provide a resource. SNs are dedicated machines which are mainly responsible for coordinating and managing the ONs.

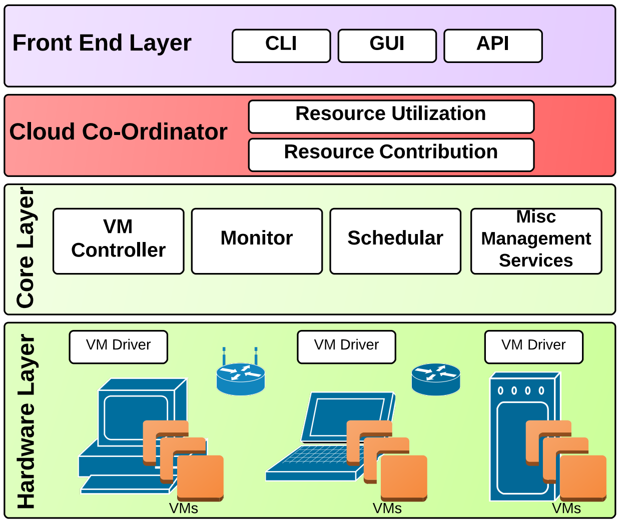

Figure 3. Architecture of the Community Cloud Manager

1) Ordinary Nodes (ONs): Each ON is assigned to a SN called *parent-SN* and holds the necessary information about it. In addition, each ON maintains locally a list called *ON_SNList* which contains the metadata of other SNs. With the information in this list, an extended registration of an ON in the system can done.

When an ON needs some resources, it sends a request to its parent-SN. Moreover, ONs periodically send a heartbeat message to their parent-SN to inform about their aliveness and inform about their current status.

2) Super Nodes (SNs): Each SN is responsible for a set of ONs and stores their metadata in a structure called *SN_ONList*. This list has to be updated after each resource sharing operation. Besides, each SN has another list called *SN_SNList* which holds the metadata of other SNs. This list is refreshed periodically to update which SN can supply how much amount of resources. For this purpose, each SN publishes their own status to other SNs, e.g. by gossiping.

3) Community Cloud Manager: Deploying community clouds in a wireless mesh network will require installing a *community cloud manager* software [6] on super nodes. Figure 3 shows the generic architecture of this community cloud manager software. The ordinary nodes form the physical layer of the cloud in a wireless mesh network. The core layer contains the software for managing and monitoring the virtual machines on ordinary nodes. The front end layer provides the interface to the infrastructure services (Infrastructure-as-a-Service, IaaS) provided by the super node.

The cloud co-ordinator connects with multiple super nodes in a mesh network to form federated super node clouds [6]. The co-ordinator is responsible for interchanging the capability and utilization of cloud resources and applies the incentive mechanism for the resource allocation.

III. INCENTIVE MECHANISM IN COMMUNITY CLOUD

Since the community networks are mainly based on voluntariness of participants, incentives to create this voluntariness is crucial to achieve the sustainability of the system [7], [8]. Our community cloud manager therefore needs to apply in resource allocation an incentive mechanism which encourages users to contribute to the system. This contribution will create a feedback loop in which increased utilization of the system would lead to interest and sustainability.

A. Requirements

We identify the following requirements for the needed incentive mechanism:

- **Decentralization:** Since a dedicated centralized repository might not be available in community cloud, incentive mechanism should be decentralized and self-managing.
- **Adaptability:** When the incentive mechanism is deployed on community network, this should not affect the principles community networks are built upon.
- **Generic:** Many services and applications may run on community clouds. The incentive mechanism should efficiently support high service diversity.
- **Rewarding:** The altruistic users which contribute resources to the cloud system should be rewarded. Effort- or contribution-based rewards are options.
- **Lightweight:** The cost of executing the incentive mechanism in terms of overhead should be low.
- **Fairness:** The incentive mechanism has to take into account the user's physical limitation with regard to the contribution made before assessing it. Therefore, users should be compared in the equity conditions.
- **Maximization of social welfare :** The incentive mechanism should focus on increasing the social welfare rather than benefiting a small set of nodes.

According the these listed requirements, we propose an effort-based incentive mechanism which applies reciprocity-based resource allocation. This mechanism is inspired by the *Parecon* economic model [9], and effort-based incentives [10]. In this model, nodes' rewards are calculated based on how much effort they put in when contributing to the system.

B. Assumptions on the physical machines

It is assumed that SNs correspond to more powerful machines which can handle the needed number of request from ONs and are enabled to monitor them. Resources can be virtual machines which are requested. ONs can be heterogeneous nodes in terms of capacity and shared sources and have asymmetric resource requests.

C. Formulations

Each user's node has a credit which reflects its contribution to the system. The credit of a node CR_i depends on the amount of resources R_i the node shared and the cost of transaction which is calculated by the time T_i during which these resources SR_i were shared. If the corresponding node shares resources, the transaction cost is added to its credit, otherwise it is subtracted.

$$CR_i = r \times R_i \mp t \times T_i \times r \times SR_i \tag{1}$$

where *r* and *t* nonzero coefficients are the factors for the shared amount of resources and the time spent for sharing, respectively.

In the effort-based incentive, the effort of a node E_i expresses its relative contribution to system, so the capacity of node C_i has to be taken into account.

$$E_i \quad = \begin{cases} \frac{CR_i}{c \times C_i} & if \quad \frac{CR_i}{c \times C_i} < 1 \\ 1 & otherwise \end{cases} \tag{2}$$

where the nonzero coefficient *c* is a factor for the capacity that the node has. As a consequence, a node with low capacity will put in more effort than a node with high capacity even if they both donate the same amount of resources to system.

The total amount of resources Ω available in the system will be equal the to sum of the available resources at each node ω_i.

$$\Omega = \sum_{i}^{all \quad nodes} \omega_i \tag{3}$$

Finally, the maximum amount of resources a node can utilize in the system is calculated depending on its effort.

$$\Delta R_i = E_i \times (\Omega - \omega_i) \tag{4}$$

D. Algorithm

The algorithm indicated in Figure 4 shows how a super node handles a request query and performs the defined mechanism. When a SN receives a request from an ON, it first checks whether the ON's credit is sufficient for the requested resource (line 1). This credit is calculated using the incentive mechanism discussed above. If it is not enough, the request is rejected. Otherwise, the query is evaluated in the *decision* function.

The *decision* function is responsible for finding the providers that are most suitable for processing the request. For this, the SN checks the available amount of resource in the *SN_ONList* (line 6). If there are not enough resources available within its ONs, it checks the local copy of other *SN_SNList* resource information (line 18). Then it forwards the query to another SN that can best satisfy the request (line 19).

If there are enough resources in the system, a *low-credit-first policy* is performed (line 8-9). This policy aims to distribute the total credit of the system among the users as fair as possible. The mechanism gives priority to the nodes that have less credit. Such nodes become resource providers and earn credit which allow them to request resources from the system in the future.

After determining which nodes will be providers, the transaction cost is added to the providers' credit (line 12) and charged from the requestor's credit (line 13). After that, the current nodes' maximum resource amount is recalculated (line 14-15).

```
Require: receive query from node $i$ with the requested amount $R_i$
 1: if $\Delta R_i >= R_i$ then
 2:     call Decision($i$, $R_i$)
 3: else
 4:     send("rejected", $i$)
 5: end if
 6: function DECISION($i$, $R_i$)
 7: if $R_i <= \Omega$ then
 8:     SN_ONList ← sort_by_credit(SN_ONList)
 9:     ProviderList[n] ← chooseProviders(SN_ONList, $R_i$)
10:     for each j in ProviderList[n] do
11:         $CostOfTransaction_{j \to i} \leftarrow rR_j + tT_j$
12:         $CR_j \leftarrow CR_j + CostOfTransaction_{j \to i}$
13:         $CR_i \leftarrow CR_i - CostOfTransaction_{j \to i}$
14:         Recalculate($\Delta R_j$)
15:         Recalculate($\Delta R_i$)
16:     end for
17: else
18:     SN ← chooseSN(SN_SNList, $R_i$)
19:     forward(SN, $i$, $R_i$)
20: end if
```

Figure 4. Handling a request query in a Super Node

IV. EVALUATION

The design goals of our proposed incentive mechanism have two success measures: *efficiency* and *fairness*. In order to evaluate them, we have implemented a simulator which simulates different policies with 250 ordinary nodes assigned to 1 super node. Different experiments are done and we analyze the results under *efficiency* and *fairness*. In each experiment the results consider the success ratio metric, which is the proportion of the successful queries to the total number of queries.

A. Efficiency

In these experiments, three different resource availabilities are created: small-resource, normal and large-resource. In the small-resource scenario, each node shares at most 25% of their own capacities, while in the large-resource scenario, this ratio is at least 75%. In the scenario denoted normal, nodes share 25%-75% of their own capacities. Overall, 12% of all resources are shared in the small-resource, 54% in the normal, and 82% in the large-resource scenario. Nodes perform 20 queries at each round over a total of 100 rounds. The requested amount of resource in these queries is at most 50% of the corresponding node's capacity.

In Figure 5, the success ratio results of the *effort-based* incentive mechanism is compared with the *contribution-based* incentive mechanism. It can be seen for both mechanisms that when the amount of available resource is increased, the success ratio also increases. Comparing the results of the two mechanisms, the success ratio obtained in the effort-based mechanism is higher.

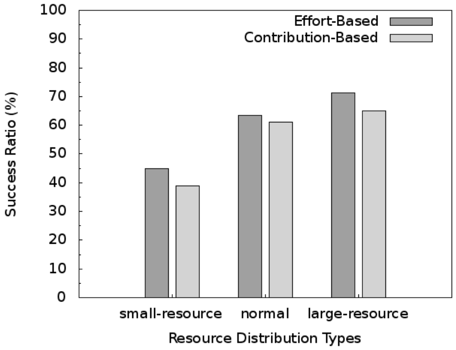

Figure 5. Overall efficiency in different Incentive Mechanisms

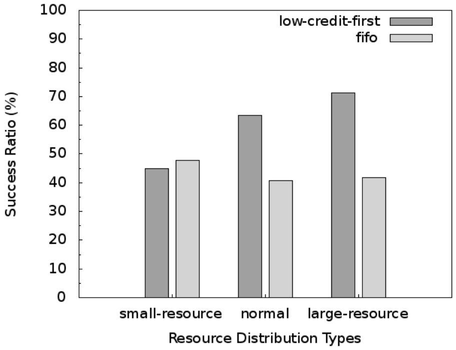

Figure 6. Overall efficiency in different Decision Mechanisms

Figure 6 compares two different decision mechanisms in the effort-based approach: *low-credit-first* which is our proposed way and *fifo* (first in first out) mechanism which is widely used as a queuing technique. It can be seen that the *low-credit-first* mechanism increases the system efficiency with increasing amount of shared data. The almost evenly distributed credits help all nodes to satisfy their requests.

B. Fairness

In these experiments, the nodes are divided equally in two classes according to their capacities: large-resource and small-resource nodes. Large-resource nodes have 4 times more capacity than small-resource nodes. All share their capacity with the system. Like previous experiments, we tested with both the effort-based and contribution-based approach.

Figure 7 shows the success ratio of both large-resource and small-resource nodes in the effort-based and contribution-based experiments along the 100 rounds. While the difference in the success ratio between large-resource (avg. 80.8%) and small-resource nodes (avg. 61.0%) in effort-based is not very large, there is a huge difference in the contribution-based mechanism between the success ratio of large-resource (avg. 61.4%) and small-resource nodes (avg. 19.0%), even though nodes share all their capacity. The reason is that the contribution-based mechanism does not take into account physical constraints of the nodes in the contributed resources. Therefore, it cannot prevent unfairness.

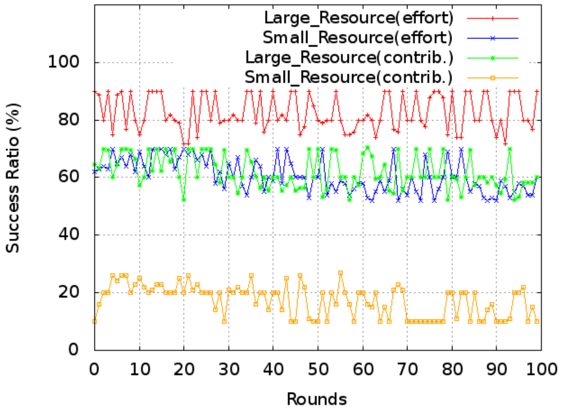

Figure 7. Success Ratio of both large-resource and small-resource nodes in different Incentive Mechanisms

The results show that allocating resources based on their effort not only ensures fairness but also improves efficiency of the system.

V. RELATED WORK

At the level of participation in community networks, reciprocal resource sharing is in fact part of the membership rules or peering agreements of many community networks. The Wireless Commons License (WCL) of many community networks states that the network participants that extend the network, e.g. contribute new nodes, will extend the network in the same WCL terms and conditions, allowing traffic of other members to transit on their own network segments.

Regarding incentive mechanisms, in the literature there are various incentive mechanisms which address the different requirements described in previous sections. None of these incentive mechanisms however target the particular situation of wireless community networks. In the following, we present a classification of the existing incentive mechanisms which are mostly designed for P2P and decentralized systems, indicating their advantages and disadvantages. In the literature, these incentive mechanisms are mainly separated into four parts [11]–[14]:

1) Inherent Generosity Schema: In this system, the percentage of free-riders and white-washing are determined based on strong mathematical formulation. If this percentage is below a threshold, the system ceases to exist as an artifact, since the number of selfish user increases unboundedly. The biggest disadvantage of this mechanism is that it is not practical.

2) Fixed-Contribution Schema: In order to participate in the network, a node has to contribute a fixed amount of resources. The nodes are centrally monitored and are required to share a minimum amount of resources. Although this mechanism has a simple design, it is centralized which is not suitable for decentralized platforms. Moreover, it suffers from free-riders.

3) Reciprocity-Based Schema: The nodes maintain historical behaviour data of other nodes and allocate the resources to them in proportion of their contribution. It can be done either in real time such as exchange-based systems like BitTorrent or non-real time such as reputation-based systems like Eigentrust [15]. Although this mechanism increases the level of cooperation, it suffers from the dependence on a third-party.

4) Monetary-Payment Schema: The users pay the resource provider for the resources. This payment issue is organized by a third-party. This schema is mainly dependent on strong economical models and micropayment systems. It requires careful handling of account management, price setting against inflation and deflation, and security issues which are not feasible for our design.

On the level of complete systems for community cloud computing [16], there are a few research prototypes that aim to provide cloud services by harvesting excess resources from machines connected via Internet. Skadsem et al. [17] provides applications for the communities by using local cloud services. Their work is similar to ours though they assume that the social mechanisms like trust in a small community do not require additional mechanisms for incentives.

The Cloud@Home [18] project has similar goals to harvest in resources from the community to meet peaks in demands. The system envisages ensuring Quality of Service (QoS) using a rewards and credit system, however the authors have not provided sufficient details to understand how these incentives will be designed. We notice that none of the found related work proposes and discusses community clouds within wireless mesh networks that form a community network.

VI. CONCLUSION

Wireless community networks would have additional value from services deployed on community clouds. A vast amount of applications could be deployed upon community clouds, boosting the usage and spread of the community network model.

We have proposed a service for reciprocal resource sharing to encourage active contributory participation of the community members to form the cloud infrastructure. This service is part of a distributed service architecture for providing cloud services that is tailored to the unique nature and conditions of community networks. Since community networks are volunteer organizations, we consider an incentive-based service essential to assure a sustainable community cloud within community networks. We have simulated the behavior of the algorithm that we propose for this service, in order to characterize how it regulates the decision on a resource request as a function of the user's resource contribution.

A next step of our work is to assess the community cloud performance. In future work we plan to run experiments to investigate the performance of such a distributed community cloud in wireless community networks.

ACKNOWLEDGEMENT

This work was supported by the European Community Framework Programme 7 FIRE Initiative projects Community Networks Testbed for the Future Internet (CONFINE), FP7-288535, and CLOMMUNITY, FP7-317879. Support is also provided by the Universitat Politècnica de Catalunya BarcelonaTECH and the Spanish Government through the Delfin project TIN2010-20140-C03-01.

REFERENCES

[1] F. A. Elianos, G. Plakia, P. A. Frangoudis, and G. C. Polyzos, "Structure and evolution of a large-scale Wireless Community Network," in *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops*. IEEE, Jun. 2009, pp. 1–6.

[2] "Guifi.net new sections of fiber deployed to the farm," 2012. [Online]. Available: http://en.wikinoticia.com/Technology/internet/122595

[3] D. Vega, L. Cerda-Alabern, L. Navarro, and R. Meseguer, "Topology patterns of a community network: Guifi.net," in *2012 1st International Workshop on Community Networks and Bottom-up-Broadband (CNBuB 2012), within IEEE WiMob*. IEEE, Oct. 2012, pp. 612–619.

[4] "Guifi.net: Open, Free and Neutral Network Internet for everybody." [Online]. Available: http://guifi.net

[5] B. Yang and H. Garcia-Molina, "Designing a super-peer network," in *Proceedings 19th International Conference on Data Engineering*. IEEE, 2003, pp. 49–60.

[6] R. Moreno-Vozmediano, R. Montero, and I. Llorente, "IaaS Cloud Architecture: From Virtualized Data Centers to Federated Cloud Infrastructures," *Computer*, 2012.

[7] G. Biczók, L. Toka, A. Gulyás, T. a. Trinh, and A. Vidács, "Incentivizing the global wireless village," *Computer Networks*, vol. 55, no. 2, pp. 439–456, Feb. 2011.

[8] M. Bina and G. Giaglis, "Unwired Collective Action: Motivations of Wireless Community Participants," in *2006 International Conference on Mobile Business*. IEEE, Dec. 2006, pp. 31–31.

[9] R. Rahman, M. Meulpolder, D. Hales, J. Pouwelse, D. Epema, and H. Sips, "Improving Efficiency and Fairness in P2P Systems with Effort-Based Incentives," *2010 IEEE International Conference on Communications*, pp. 1–5, May 2010.

[10] D. Vega, R. Meseguer, F. Freitag, and S. Ochoa, "Effort-Based Incentives for Resource Sharing in Collaborative Volunteer Applications," in *2013 17th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2013)*. IEEE, 2013.

[11] K. Zhang, N. Antonopoulos, and Z. Mahmood, "A taxonomy of incentive mechanisms in peer-to-peer systems: Design requirements and classification," *International Journal on Advances in Networks and Services*, vol. 3, no. 1, pp. 196–205, 2010.

[12] M. Babaioff, J. Chuang, and M. Feldman, "Incentives in peer-to-peer systems," in *Algorithmic Game Theory*. Cambridge University Press, 2007, pp. 593–612.

[13] M. Feldman and J. Chuang, "Overcoming free-riding behavior in peer-to-peer systems," *ACM SIGecom Exchanges*, vol. 5, no. 4, pp. 41–50, Jul. 2005.

[14] Y. Tang, H. Wang, and W. Dou, "Trust based incentive in P2P network," in *IEEE International Conference on E-Commerce Technology for Dynamic E-Business*, no. 90104020. IEEE Comput. Soc, 2004, pp. 302–305.

[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the twelfth international conference on World Wide Web - WWW '03*. New York, New York, USA: ACM Press, 2003, p. 640.

[16] A. Marinos and G. Briscoe, "Community Cloud Computing," *Computing*, vol. 5931, no. December, p. 11, Jul. 2009.

[17] M. K. Skadsem, R. Karlsen, G. Blair, and K. Mitchell, "Community Cloud - Cloud Computing for the Community," in *Proceedings of the 1st International Conference on Cloud Computing and Services Science*. Setubal, Portugal: SciTePress, 2011, pp. 418–423.

[18] S. Distefano and A. Puliafito, "Cloud@Home: Toward a Volunteer Cloud," *IT Professional*, vol. 14, no. 1, pp. 27–31, Jan. 2012.