Towards Incentive-based Resource Assignment and Regulation in Clouds for Community Networks

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

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

Abstract. Community networks are built with off-the-shelf communication equipment aiming to satisfy a community's demand for Internet access and services. These networks are a real world example of a collective that shares ICT resources. But while these community networks successfully achieve the IP connectivity over the shared network infrastructure, the deployment of applications inside of community networks is surprisingly low. Given that community networks are driven by volunteers, we believe that bringing in incentive-based mechanisms for service and application deployments in community networks will help in unlocking its true potential. We investigate in this paper such mechanisms to steer user contributions, in order to provide cloud services from within community networks. From the analysis of the community network's topology, we derive two scenarios of community clouds, the local cloud and the federated cloud. We develop an architecture tailored to community networks which integrates the incentive mechanism we propose. In simulations of large scale community cloud scenarios we study the behaviour of the incentive mechanism in different configurations, where slices of homogeneous virtual machine instances are shared. Our simulation results allow us to understand better how to configure such an incentive mechanism in a future prototype of a real community cloud system, which ultimately should lead to realisation of clouds in community networks.

Keywords: incentive mechanisms, cloud computing, community networks, distributed resource sharing

1 Introduction

Community networks aim to satisfy a community's demand for Internet access and services using open unlicensed wireless spectrum and off-the-shelf communication equipment. Most community networks originated in rural areas which commercial telecommunication operators left behind when focusing the deployment of their infrastructure on urban areas. The lack of broadband access brought together different stakeholders of such geographic areas to team up and invest, create and run a community network as an open telecommunication infrastructure based on self-service and self-management by the users [1].

These community networks are a real world example of a collective that shares information and communication technology (ICT) infrastructure and human resources. The ICT resources shared are the bandwidth of the wireless network formed by the networking hardware belonging to multiple owners. This bandwidth allows members of the community network obtaining access to the Internet or use services and applications inside of the community network. The human resources shared are the time and knowledge of the participants, needed to maintain the network and technically organize it for further growth.

Sharing of network bandwidth has early been identified as essential and is part of the membership rules or peering agreements of many community networks, which regulate the usage and growth of the network. The Wireless Commons License (WCL) [2] 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. Since this sharing is done by all members, community networks successfully operate as IP networks.

Today's Internet, however, is more than bandwidth resources. Computing and storage resources are shared through Cloud Computing, offering virtual machine instances over infrastructure services, APIs and support services through platform-as-a-service, and Web-based applications to end users through software-as-a-service. These services, now common practice in today's Internet, hardly exist in community networks [3]. Services offered in community networks still run on machines exclusively dedicated to a single member. Community network members, however, do use commercial cloud solutions, for instance for network administration, where sometimes a commercial storage service is used for node data. Why have clouds not emerged inside of the community networks?

We argue that community cloud, a cloud infrastructure formed by community-owned computing and communication resources, has many technical and social challenges so that the main drivers of today's contribution to community networks, voluntariness and altruistic behaviour, are not enough to successfully cope with it. Our hypothesis is that for community cloud to happen, the members' technical and human contribution needed for such a cloud, needs to be steered by incentive mechanisms that pay back the users' contribution with a better quality of experience for them.

In this paper, we present an incentive mechanism tailored to community networks. The main contributions of this paper are the following:

1. From the analysis of the key socio-technical characteristics of community networks, we identify two scenarios for community clouds, the local clouds and federated clouds, for which a community cloud management system is proposed.
2. We design an incentive mechanism that is part of the community cloud architecture and evaluate its behaviour in simulations of community cloud scenarios.

We elaborate our contributions in the following way: In section 2 we present our system model and design. In section 3, we evaluate our incentive mechanism in a community cloud scenario. In section 4 we relate the work of other authors with our results. We discuss open issues in section 5 on future work and in section 6 we conclude our findings.

2 System Model and Design

Our incentive mechanism for community cloud targets real community networks so it must be integrated into an architecture, design and implementation which fits into these conditions and scenarios. In this section, we first analyse the topology of community networks from which we develop two main cloud scenarios we foresee for them. We then present the conceptual overview of a cloud management system suitable for community networks, of which we identify the resource assignment and regulation mechanism as a key component.

2.1 Topology of Community Networks

The community network generally has two different types of nodes, super nodes (SN) and ordinary nodes (ON). Super nodes have at least two wireless links, each to other super nodes. Most super nodes are installed in the community network participant's premises. A few super nodes, however are placed strategically on third party location, e.g. telecommunication installations of municipalities, to improve the community network's backbone. Ordinary nodes only connect to a super node, but do not route any traffic. A topological analysis of the Guifi.net community network [4] indicates that from approximately 17,000 analysed nodes of Guifi.net, 7% are super nodes while the others are ordinary nodes.

2.2 Community Cloud Scenarios

The scenario of *local community cloud* is derived from the topology of community network and the observed characteristics of the strength of the social network within community network zones. In the local community cloud, a super node is responsible for the management of a set of attached nodes contributing cloud resources. From the perspective of the attached nodes, this super node acts as a centralized unit to manage the cloud services.

Multiple super nodes in a community network can connect and form *federated community clouds* [5]. The super node connects physically with other super nodes through wireless links and logically in an overlay network to other SNs that manage local clouds. SNs coordinate among each other and the requests originating from one SN's zone can therefore be satisfied by the resources allocated from another SN's zone.

2.3 Community Cloud Manager

The option we foresee for enabling a cloud in a community network is deploying a cloud management system tailored to community networks on a super node. We propose a conceptual overview for such a system in Figure 1 which consists of the following.

- The ordinary nodes of the community network provide hardware resources isolated as virtual machine (VM) instances and form the hardware layer of the cloud architecture.
- The core layer residing in the super node contains the software for managing the virtual machines on ordinary nodes.
- The cloud coordinator is responsible for the federation of the cloud resources which are independently managed by different local community clouds. The cloud coordinator components in different SNs connect with each other in a decentralized manner to exchange relevant information about managing the available resources.
- The front end layer provides the interface for accessing resources from the cloud as Infrastructure-as-a-Service (IaaS).

The core of cloud management system is virtual machine manager that is responsible for instantiating, scheduling and monitoring virtual machines on the nodes. There are some cloud management systems available to manage public and private clouds, for example OpenNebula [5] and OpenStack [6] are among the most consolidated and popular open source tools. Such cloud management systems are then tailored for community networks by extending them with implementing the cloud coordinator and its services on top of them, to address the particular conditions of community networks.

2.4 Incentive Mechanisms in Community Cloud

Participants in a community network are mainly volunteers that act independently and are not obliged to contribute. To ensure sustainability and growth of the community cloud, incentive mechanisms are needed that encourage members to contribute with their hardware, effort and time [7, 8]. When designing such mechanisms, the heterogeneity of the nodes and communication links has to be considered since each member brings in a widely varying set of resources and physical capacity to the system.

Most peer-to-peer (P2P) systems implement incentive mechanisms based on contribution where nodes are rewarded according to resources they donate to the system [9]. We suggest an effort-based incentive mechanism for community cloud where effort is defined as contribution relative to the capacity of a node [10]. This mechanism is inspired by the *Parecon* economic model [11–13] which focuses on social welfare by considering inequality among nodes. Nodes with different capacity cannot have same contribution to the system but in this mechanism they get same reward if they share as much as possible of their capacity as we explain in the following.

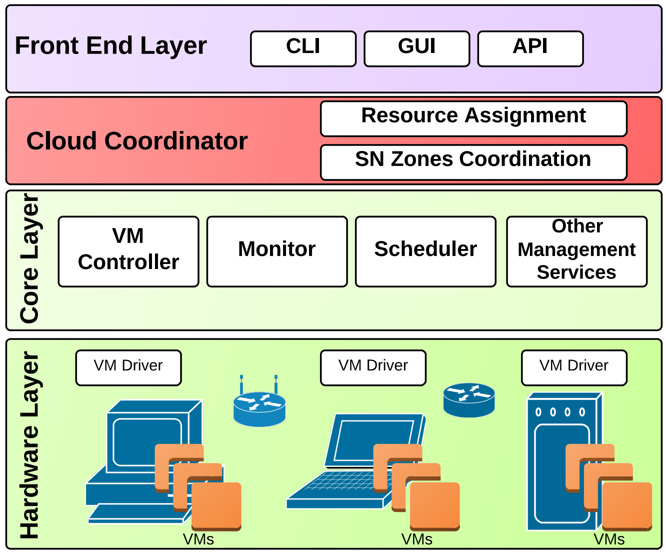

Fig. 1. Conceptual overview of the Community Cloud Manager

Formulations We first discuss here the criteria that a super node uses to evaluate requests from ordinary nodes. When a node asks for a resource from a SN, which in this case means to commit an instance of virtual machine for a given duration, the SN first checks whether the ON's credit is sufficient to cover the cost of the transaction. The cost is proportional to the number of resources requested R_i and the duration T_i for how long they are required.

$$transaction\_cost = \gamma R_i \times \rho T_i \tag{1}$$

where γ and ρ are nonzero coefficients for the amount and duration of resources shared respectively.

If the requesting node does not have enough credit, the request is rejected. Otherwise, the SN searches for nodes that have resources available. It selects as many nodes as possible from its local zone as providers. If the demand cannot be met locally, the SN forwards the request to super nodes in the federated community cloud.

Now we consider how the SN manages the credits of the nodes that take part in the transaction. For each node which contributed its resources to fulfil the request, the SN calculates the transaction cost as shown above and adds it to that node's credits. The cost is deducted from the credits of the node that consumed the resources. After the transaction is completed, the effort for each node involved in the transaction is recalculated as in [10] by:

$$E_i = \begin{cases} \frac{credit_i}{\epsilon C_i} & if \quad \frac{credit_i}{\epsilon C_i} < 1 \\ 1 & otherwise \end{cases} \tag{2}$$

```
Require: receive query from node i with the requested amount R_i and the time T_i
 1: calculate(ΔR_i)
 2: if R_i <= ΔR_i then
 3:    call Decision(i, R_i, T_i)
 4: else
 5:    send("rejected", i)
 6: end if
 7: function DECISION(i, R_i, T_i)
 8: if R_i <= Ω then
 9:    ProvidersList[n] ← high_score_first(ON_List, R_i)
10:    for each j in ProviderList[n] do
11:       CostOfTransaction_{j→i} ← R_j^r * T_j^t
12:       update_credits(CostOfTransaction_{j→i})
13:       update_database(ON_List)
14:    end for
15: else
16:    SN ← low_credit_first(SN_List,R_i, reserved_ratio)
17:    forward(SN,i, R_i, T_i )
18: end if
```

Fig. 2. Algorithm for handling requests from ordinary nodes

where ϵ is nonzero coefficient for the capacity of the node. The effort of a node expresses its relative contribution to the system, since the mechanism considers the capacity C_i of a node as well. This means that a node with low capacity puts in more effort than a node with high capacity if they both donate same amount of resources to the system.

The total amount of resources available Ω in the system is sum of the resources ω_i shared by each node.

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

And the maximum resource ΔR_i a node can consume depends on its effort.

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

Algorithm for Requests Processing Figure 2 shows algorithm for how a SN handles request from a node in its zone. When SN receives request, it first calculates that node's allowance ΔR_i to confirm whether it has enough credit to fulfil the request. If not, the request is rejected, otherwise the algorithm calls *decision* function which searches for available resources (lines 1–5).

The *decision* function first checks if enough resources are available in the local zone (line 8), and selects the nodes that will provide the resources from its local zone using *high-score-first* policy (line 9). The idea is to give preference to the nodes that need credit the most for participating in the system. If SN cannot satisfy request from its local nodes, it forwards request to one of its neighbouring super nodes which is chosen using *low-credit-first* policy (lines 16–18). This allows the zone with depleted credits to earn more so its nodes can be active the system again. After the provider nodes commit resources, SN calculates cost of the transaction and updates the nodes' credits, deducting credits from the requester and increasing credits of the providers (lines 10–14).

Policies for Nodes Selection When SN processes requests for resources, there may be multiple nodes that can be providers so SN applies a selection policy for prioritizing which nodes to choose. Similarly when SN forwards requests to other SN zones, it also has to select between multiple zones that have resources available. We evaluated a number of selection criteria that can be employed in above algorithm, and observed in experiments that *low-credit-first* and *high-score-first* policies were better in terms of efficiency of the system. In the following we explain these different policies and discuss the motivation behind them.

- **Low Credit First Selection.** When nodes consume resources, their credit gets spent and with time their credit may be too low to request any resources. Such nodes can provide their resources to other nodes and earn credit allowing them to participate in the system again. This policy gives priority to nodes with low credit with the aim to ensure that most nodes participate in the system and are not left out because of lack of credit.
  When multiple SN zones participate in the system, same problem exists since nodes in a particular zone may have all spent their credit and cannot request any more resources. So the algorithm above gives preference to such zones by applying low-credit-first policy when selecting other SNs to forward requests.
- **High Score First Selection.** One issue with the low-credit-first approach is that it does not differentiate among nodes with low credit. Some of the nodes may be inactive and not making any requests while others may be getting their requests rejected because of inadequate credit. In this policy, the SN tracks unsuccessful attempts by each node and assigns it a *score* calculated as follows. Nodes with higher score get preference so they can recover their credit.

$$score_i = \frac{attempts_i}{credit_i} \tag{5}$$

- **Other Policies.** We also considered following policies and compared their effect on efficiency of the system.
  - First-in-first-out (FIFO). In this simple policy, as soon as nodes have free resources, they register their availability with SN which keeps on adding them in a queue. When processing requests, the SN selects a node that has been in the queue the longest.
  - Random. In this policy, SN picks a node at random from the queue.
  - High credit first. This is the opposite of low-credit-first policy and here nodes with more credits are chosen first.

Table 1. Configuration for each node in a zone with shared and total instances

Node Behaviour	Shared	Small capacity	Medium capacity	Large capacity
Selfish	33%	ON1 (1/3)	ON2 (2/6)	ON3 (3/9)
Normal	66%	ON4 (2/3)	ON5 (4/6)	ON6 (6/9)
Altruistic	100%	ON7 (3/3)	ON8 (6/6)	ON9 (9/9)

3 Evaluation

In the past work [10], we studied incentive mechanisms for resource regulation within a single SN zone which corresponds to local community cloud scenario. Here we extend our simulator to study resource regulation across multiple SN zones covering both local and federated community cloud scenarios. In addition to simulations, we also implemented and deployed a prototype of the regulation component of Cloud Coordinator on nodes of a real community network using the Community-Lab testbed [14] provided by the CONFINE project [15]. However, as only a handful of nodes are made available currently, the analysis of our proposed system on greater scale using the real prototype system is too limited. Therefore, we focus here on reporting results from the simulation experiments, where our scenario could be extended to a community cloud consisting of 1,000 nodes.

3.1 Experiment Setup

We simulate a community network comprising of 1,000 nodes which is divided into 100 zones and each zone has one super node and nine ordinary nodes. The zones are distributed in a small world topology where each zone is neighbour to 10 other zones. This approximation holds well for real world community networks as, for example, topology analysis of Guifi.net [4] shows that the ratio of super node to ordinary nodes is approximately 1 to 10. Each ordinary node in the simulation can host a number of VM instances that allows users' applications to run in isolation. Nodes in the zone have two main attributes, one is capacity which is the number of available VM instances, and other is sharing behaviour which is how many instances are shared with other nodes. Table 1 shows the different configurations for each of the nine ONs in each zone. Nodes with low, medium and high capacity host 3, 6 and 9 VM instances respectively and they exhibit selfish, normal or altruistic behaviour sharing one-third, two-thirds or all of their VM instances. For example, node ON2 has medium capacity with 6 instances and exhibits selfish behaviour reserving 4 instances for itself and contributing only 2 to the system.

When the experiment runs, nodes make requests for resources proportional to their capacity asking for two-thirds of their capacity. For instance nodes with capacity of 3, 6 and 9 VM instances request 2, 4 and 6 instances respectively.

Table 2. Success ration of nodes for different configurations with effort and contribution based incentives

Node Behaviour	Incentives	Small capacity	Medium capacity	Large capacity
Selfish	effort-based	54%	53%	50%
	contribution-based	66%	59%	39%
Normal	effort-based	90%	91%	86%
	contribution-based	97%	77%	66%
Altruistic	effort-based	97%	94%	86%
	contribution-based	97%	85%	65%

Nodes request instances for fixed duration and after transaction is complete wait briefly before making further requests.

3.2 Experimental Results

We evaluate the impact of the effort-based incentive mechanisms in the system in simulation experiments and discuss the results below. We study the success ratio, i.e. number of requests fulfilled versus total requests, and the overall resource utilization in the system.

Ratio of Successful Requests Table 2 shows the success ratio for requests made by different nodes analysed both with the effort-based and contribution-based incentive mechanisms. We first notice that the success ratio values decrease as the capacity of the nodes increases. This is explained by the fact that nodes with greater capacity request more instances and so have a higher chance getting rejected either because there are not many resources available in the system or because the requesting nodes do not have sufficient credit.

Moreover, when we compare success ratio for nodes as capacity increases, we observe greater variation in the case of contribution-based incentives. For instance, for the normal sharing behaviour the values range from 66% to 97% for contribution-based incentives, but from 86% to 90% for effort-based incentives. This is explained by the fact that contribution-based approach does not take heterogeneity of nodes into account and penalizes nodes with low capacity as they cannot contribute as much to the system as others. These results indicate that effort-based incentives ensure *fairness* in the system since the nodes with the same sharing behaviour are treated equally irrespective of their capacity.

Breakdown of Request Responses Figure 3 shows the breakdown of successful and rejected requests. The success ratio is higher for effort-based incentives. Moreover, contribution-based mechanism has greater share of requests rejected because of lack of credit. This indicates that effort-based incentives result in

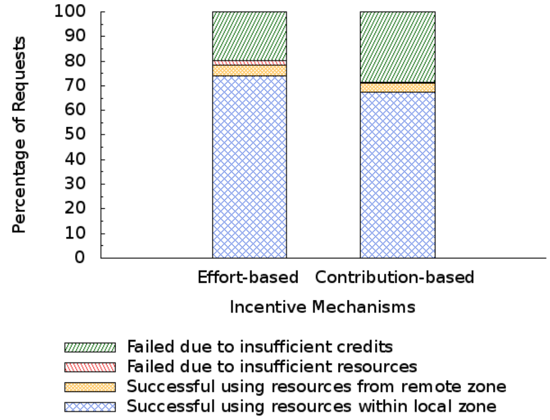

Fig. 3. Breakdown of outcome of requests with effort and contribution based mechanisms

better efficiency as more resources remain utilized. Another observation is that majority of requests are fulfilled using resources from local zone with very few requests forwarded to other zones.

Resource Utilization Figure 4 shows the proportion of resources utilized in the system along the execution of a 24 minutes experiment for effort and contribution based approach. In the start all nodes have enough credit and the resource utilization is high. Then it drops to below 60% at around the 12^{th} minute. Then, since most of the nodes completed their transactions and consumed their credits, the utilization decreases significantly. The effort-based approach though achieve a higher resource utilization during that time.

Nodes Selection Policies Figure 5 shows the effect of different node selection policies on the success ratio when using effort-based incentives. High-credit-first and first-in-first-out policies perform poorly since they do not consider the credits of the nodes and so fail in ensuring a balanced distribution across the system. The low-credit-first and high-credit-first policies perform better since they give preference to nodes with low credit allowing them to earn more so that they can be successful with their future requests.

4 Related Work

After the prevalence of public clouds [16], there is now increasing interest in providing cloud services by harvesting excess resources from the idle machines connected to the Internet [17]. Having different service level requirements and conditions, different solutions for how resources are contributed to build clouds

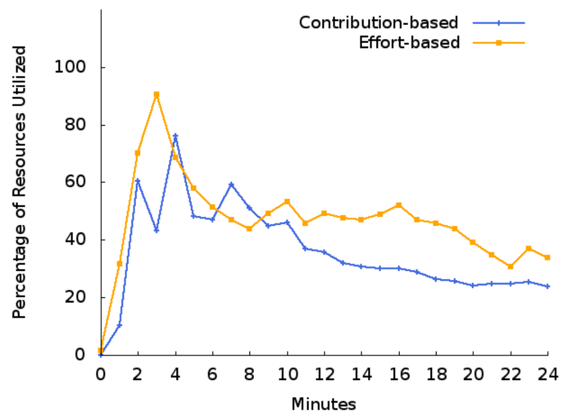

Fig. 4. Resource utilization along 24 minutes of the experiment

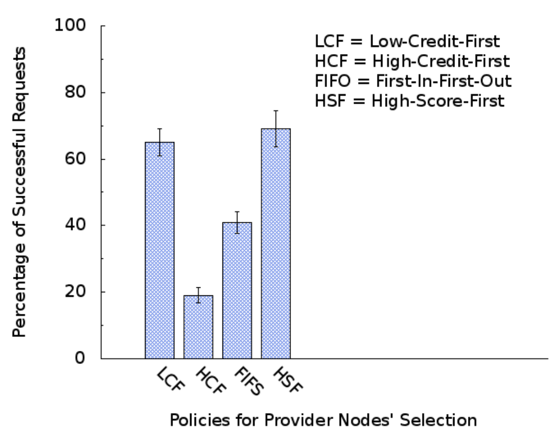

Fig. 5. Success ratio comparison of provider ON selection strategies

have been found. Commercial clouds have dedicated resources that are financed by the users who pay in hard currency to use the cloud services. Previous distributed multi-owned computing platforms like Seti@Home [18], HTCondor [19] and Seattle [20] have relied on altruistic contribution of volunteer users. PlanetLab [21] requires for granting resource usage a prior fixed contribution before the services are made available. None of these cases, however, correspond to the concrete situation of community networks. In order to build a cloud platform within a community network, there is a need to create incentives to encourage active participation from the members of the community.

Various incentive mechanisms have been studied for P2P and decentralized systems that address different requirements for ensuring a sustainable volunteer-based system [9]. P2P systems like BitTorrent [22] incentivize using reciprocity based schemes where users consume resources in proportion to their contribution. Most of these schemes do not take heterogeneity and varying capacity of different nodes into account so nodes with limited capacity are at a disadvantage because they do not benefit as much from the system even though they may be actively contributing to the system. Recent work in cloud systems have also employed similar reciprocity based schemes, for example, Cloud@Home project [23] envisages ensuring Quality of Service (QoS) using a rewards and credit system. Fixed contribution schemes [21] need centralized management which are not suitable and scalable for decentralized systems like community networks. Monetary based schemes [24–27] are founded on economic models and need careful micro-management which makes it complicated to implement for a large decentralized system like community networks.

Regarding the different incentive schemes, our approach takes advantage of elements of the monetary-payment scheme, in the sense that credits are used to reflect the interchange of resources between consumers and providers. These credits are part of the components of the incentive mechanism that we propose for community clouds. We notice that none of the found related work focus on wireless community networks such as targeted by us.

5 Future Work

We have investigated incentive mechanisms for community clouds based on reciprocal resource sharing. Our results indicate their impact on the efficiency of the system and on regulating the resource assignments. The understanding gained from the different experimental results helps in the design of the policies that such incentive mechanism could follow in a future prototype of real community cloud system.

Our results, however, have revealed new issues that are to be addressed in the next steps towards a real cloud system. First, we have not yet investigated the behaviour of the incentive mechanism for extended periods of time. Further experiments are needed to study how the mechanism can be used for long durations. Secondly, we have not yet investigated the incentive mechanism in a prototype deployed in a real community network.

For the permanent operation of the cloud system with the incentive mechanism, the mechanism needs to be able to adapt to the system state in runtime. The mechanism will need to be able to take into account the evolution of the system with regards to users, resources, and different kind of behaviours. Therefore, parameters of the incentive mechanism will need to be defined as functions of the system state in order to account and decide correctly on the current situation. In order to further develop this runtime adaptability, a two-fold approach, which on one hand extends the simulations with refined system models and on the other hand evaluates the performance of deployed prototype components, is suggested to assure the realisation of an operative adaptive system.

A prototype of the incentive mechanism integrated in a cloud management platform is needed to be able to obtain performance results from real users and services. An operative modular system is needed that allows an easy modification of its components according to the simulation results. The transfer of the simulation results to the deployed system should be required, in order to assure that the simulated system model reflects the real system, and that the obtained findings can actually be brought into the real system in a feasible way.

Finally, the deployment of several federated clouds with real users and real usage should ultimately be undertaken. Such large-scale cloud deployments need to have an extended implementation of a communication middleware for the coordination in a network of super nodes, complemented by additional services, to fully achieve an incentive-based resource assignment. For such systems, additional work is needed to develop in detail the feedback loop between the user's contribution and the experience the user obtains from the cloud services, needed for the building and maintenance of a cloud in community networks.

6 Conclusion

Community clouds are motivated by the additional value they would bring to community networks. Deploying applications in community clouds will boost the usage and spread of the community network model as ICT infrastructure for society. This paper builds upon the topology of community networks to derive two community cloud scenarios, local community cloud and federated community cloud. A community cloud architecture is then proposed which fits into these scenarios. The need for an incentive mechanism in order to community clouds to happen is stated, since for the contribution of any resources the motivation of the users is needed. This incentive mechanism is specified and implemented in a simulator in order to be able to perform assessments for large scale scenarios. With simulation experiments we characterized the behaviour of different settings of the incentive mechanism and evaluated the success ratio of nodes and resource utilization. A deeper analysis of the behaviour allowed us to better understand the influence of the different configuration options. The incentive mechanism has been designed and evaluated taking into account the conditions of community networks. Therefore, we expect our results to be transferable to a prototype of a real community cloud system.

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. Elianos, F.A., Plakia, G., Frangoudis, P.A., Polyzos, G.C.: 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 (June 2009)
2. Guifi.net: Commons for Open, Free & Neutral Network (OFNN). `http://guifi.net/es/ProcomunXOLN` (2010)
3. Khan, A.M., Sharifi, L., Veiga, L., Navarro, L.: Clouds of Small Things: Provisioning Infrastructure-as-a-Service from within Community Networks. In: 2nd International Workshop on Community Networks and Bottom-up-Broadband (CNBuB 2013), within IEEE WiMob, Lyon, France (October 2013)
4. Vega, D., Cerda-Alabern, L., Navarro, L., Meseguer, R.: 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 (October 2012) 612–619
5. Moreno-Vozmediano, R., Montero, R.S., Llorente, I.M.: IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures. Computer **45**(12) (December 2012) 65–72
6. OpenStack: Open Source Cloud Computing Software. http://www.openstack.org
7. Biczók, G., Toka, L., Gulyás, A., Trinh, T.a., Vidács, A.: Incentivizing the global wireless village. Computer Networks **55**(2) (February 2011) 439–456
8. Bina, M., Giaglis, G.: Unwired Collective Action: Motivations of Wireless Community Participants. In: 2006 International Conference on Mobile Business, IEEE (December 2006)
9. Zhang, K., Antonopoulos, N., Mahmood, Z.: A taxonomy of incentive mechanisms in peer-to-peer systems: Design requirements and classification. International Journal on Advances in Networks and Services **3**(1) (2010) 196–205
10. Buyuksahin, U., Khan, A., Freitag, F.: Support Service for Reciprocal Computational Resource Sharing in Wireless Community Networks. In: 5th International Workshop on Hot Topics in Mesh Networking (IEEE HotMESH 2013), within IEEE WoWMoM, Madrid, Spain (June 2013)
11. Rahman, R., Meulpolder, M., Hales, D., Pouwelse, J., Epema, D., Sips, H.: Improving Efficiency and Fairness in P2P Systems with Effort-Based Incentives. 2010 IEEE International Conference on Communications (May 2010) 1–5
12. Vega, D., Meseguer, R., Freitag, F., Ochoa, S.: 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 (June 2013)

13. Vega, D., Messeguer, R., Ochoa, S.F., Freitag, F.: Sharing hardware resources in heterogeneous computer-supported collaboration scenarios. Integrated Computer-Aided Engineering **20**(1) (2013) 59–77
14. Neumann, A., Vilata, I., Leon, X., Escrich, P., Navarro, L., Lopez, E.: Community-Lab: Architecture of a Community Networking Testbed for the Future Internet . In: 2012 1st International Workshop on Community Networks and Bottom-up-Broadband (CNBuB 2012), within IEEE WiMob, IEEE (October 2012)
15. Braem, B., Baig Viñas, R., Kaplan, A.L., Neumann, A., Vilata i Balaguer, I., Tatum, B., Matson, M., Blondia, C., Barz, C., Rogge, H., Freitag, F., Navarro, L., Bonicioli, J., Papathanasiou, S., Escrich, P.: A case for research with and on community networks. ACM SIGCOMM Computer Communication Review **43**(3) (July 2013) 68–73
16. Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A.: A view of cloud computing. Communications of the ACM **53**(4) (April 2010) 50
17. Marinos, A., Briscoe, G.: Community Cloud Computing. Cloud Computing **5931**(December) (July 2009) 472–484
18. Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D.: SETI@home: an experiment in public-resource computing. Communications of the ACM **45**(11) (November 2002) 56–61
19. Thain, D., Tannenbaum, T., Livny, M.: Distributed computing in practice: the Condor experience. Concurrency and Computation: Practice and Experience **17**(2-4) (February 2005) 323–356
20. Cappos, J., Beschastnikh, I., Krishnamurthy, A., Anderson, T.: Seattle: a platform for educational cloud computing. In: Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09), New York, New York, USA, ACM Press (2009) 111–115
21. Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., Bowman, M.: PlanetLab: An Overlay Testbed for Broad-Coverage Services. ACM SIGCOMM Computer Communication Review **33**(3) (July 2003) 3
22. Cohen, B.: Incentives build robustness in BitTorrent. Workshop on Economics of Peer-to-Peer systems **6** (2003) 68–72
23. Distefano, S., Puliafito, A.: Cloud@Home: Toward a Volunteer Cloud. IT Professional **14**(1) (January 2012) 27–31
24. Punceva, M., Rodero, I., Parashar, M., Rana, O.F., Petri, I.: Incentivising resource sharing in social clouds. Concurrency and Computation: Practice and Experience (March 2013)
25. Roovers, J., Vanmechelen, K., Broeckhove, J.: A Reverse Auction Market for Cloud Resources. In Vanmechelen, K., Altmann, J., Rana, O.F., eds.: GECON 2011. Volume 7150 of LNCS. Springer Berlin Heidelberg (2012) 32–45
26. Petri, I., Rana, O., Cosmin Silaghi, G.: SLA as a Complementary Currency in Peer-2-Peer Markets. In Altmann, J., Rana, O., eds.: GECON 2010. Volume 6296 of LNCS. Springer Berlin Heidelberg (2010) 141–152
27. Toka, L., Maill, P.: Managing a Peer-to-Peer Backup System: Does Imposed Fairness Socially Outperform a Revenue-Driven Monopoly? In Veit, D., Altmann, J., eds.: GECON 2007. Volume 4685 of LNCS. Springer Berlin Heidelberg (2007) 150–163