

Prototyping Incentive-based Resource Assignment for Clouds in Community Networks

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

Department of Computer Architecture
Universitat Politècnica de Catalunya
Barcelona, Spain

Email: {mkhan, ubuyuksa, felix}@ac.upc.edu

Abstract—Wireless community networks are a successful example of a collective where communities operate ICT infrastructure and provide IP connectivity based on the principle of reciprocal resource sharing of network bandwidth. This sharing, however, has not extended to computing and storage resources, resulting in very few applications and services which are currently deployed within community networks. Cloud computing, as in today’s Internet, has made it common to consume resources provided by public clouds providers, but such cloud infrastructures have not materialized within community networks. We analyse in this paper socio-technical characteristics of community networks in order to derive scenarios for community clouds. Based on an architecture for such a community cloud, we implement a prototype for the incentive-driven resource assignment component, deploy it in a testbed of community network nodes, and evaluate its behaviour experimentally. Our evaluation gives insight into how the deployed prototype components regulate the consumption of cloud resources taking into account the users’ contributions, and how this regulation affects the system usage. Our results suggest a further integration of this regulation component into current cloud management platforms in order to open them up for the operation of an ecosystem of community cloud.

Index Terms—cloud computing; community networks; incentive mechanisms; resource allocation

I. INTRODUCTION

Community networking is a shared communication infrastructure in which citizens build and own open communication networks. Most of these community networks are based on Wi-Fi technology such as ad-hoc networks or IEEE 802.11a/b/g/n access points in the first hop and long-distance point-to-point Wi-Fi links for the trunk network. Recently, a growing number of optical fibre links are also being deployed [1]. Despite the lack of reliable statistics, community networks seem to be rather successful. There are several large community networks in Europe, having from 500 to 20,000 nodes, such as FunkFeuer¹, AWMN², Guifi.net³, Freifunk⁴, and many others worldwide. Figure 1 shows the wireless links and nodes of the Guifi.net community network in the area of Barcelona.

The community cloud we present in this paper is the vision of a cloud deployment in community networks [2],

[3]: A cloud hosted on community-owned computing and communication resources providing services of local interest. The concept of community clouds has been introduced in its generic form before, e.g. [4], [5], as a cloud deployment model in which a cloud infrastructure is built and provisioned for an exclusive use by a specific community of consumers with shared concerns and interests, owned and managed by the community or by a third party or a combination of both.

Community networks successfully operate as IP networks, since the nodes’ bandwidth is shared among all the members in a reciprocal manner. While there are also services offered from within the community networks, most members of community networks use the infrastructure solely to access the Internet, and they consume services in the Internet and not within the community network. If there are services inside the network, they usually run on machines exclusively used by a single member (normally the owner of the machine). We emphasize that the sharing of storage and computational resources, which is now common practice in today’s Internet through cloud computing, hardly exists in community networks.

Community networks are an ecosystem which is able to regulate and maintain itself, some of the community networks are there for even more than a decade. Participants of the community network not only contribute infrastructure to the network, but also their knowledge, time and effort for successful operation of the network. We anticipate that cloud infrastructures for community networks will need additional incentive mechanisms in order to achieve sustainability. In this paper we study an incentive-mechanism for clouds in community networks, keeping in view key characteristics of community networks and the scenarios we foresee for community clouds. Our approach is to do the evaluation with a prototype, which will allow us to derive additional conclusions regarding its feasibility for implementation and deployment on a wider scale.

The main contributions of this paper are the following:

- 1) We identify a community cloud scenario, envisioned as a federation of local clouds, which is derived from a socio-technical analysis of community networks.
- 2) We implement a proposed incentive mechanism in a regulation component and deploy it in real nodes of a community network.

¹<http://funkfeuer.at>

²<http://awmn.gr>

³<http://guifi.net>

⁴<http://freifunk.net>

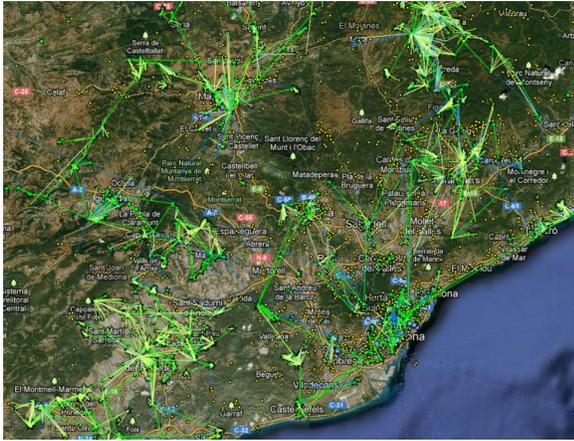


Figure 1. Guifi.net nodes and links in the area around Barcelona

- 3) We evaluate experimentally with the deployed prototype the behaviour of the incentive-driven resource assignment in the community cloud scenario.

We elaborate our contributions in the following way. In section II, we analyse community networks and bring about the community cloud scenario. In section III, we discuss a cloud architecture applicable to the topology of community network deployments. In section IV, we introduce the prototype implementation for the resource assignment component of the community cloud architecture. In section V, we evaluate in experiments the resource assignment behaviour of the prototype by deploying it in a testbed of real community network nodes. In section VI, we relate to the work of other authors, and in section VII, we conclude and indicate future work.

II. CLOUDS IN COMMUNITY NETWORKS

Since our community cloud aims to be used in real community networks, it is a must that our architecture, design, implementation and deployment fits into these conditions and scenarios. We focus our analysis on the Guifi.net community network, which is considered the largest community network worldwide, and it is where we have also deployed our prototype.

A. Community Networks

1) *Nodes and topological aspects of community networks:* A community network distinguishes between super nodes (SN) and ordinary nodes (ON). SNs have at least two wireless links, each to other SNs. Most SNs are installed in the community network participant's premises. A few SNs, however, are placed strategically on third party location, e.g. telecommunication installations of municipalities, to improve the community network's backbone. ONs only connect to a SN, but do not route any traffic. A topological analysis of the Guifi.net community network [6] indicates that from approximately 17,000 analysed nodes of Guifi.net, 7% are SNs while the others are ONs.

Figure 2 shows the outdoor view of a community network SN. The equipment, mainly antennas and radios, is used for



Figure 2. Super node with outdoor hardware for wireless links between community nodes



Figure 3. Indoor hardware of a community network node with router, server and cloud resource

building wireless links between other SNs. Figure 3 shows an example of the indoor hardware of a SN. The router used is a Microtik RB750, while a Jetway JBC362F36W with Intel Atom N2600 CPU, 2GB RAM and 64GB USB has been added to become a cloud resource for the community cloud. A laptop is used as an additional server, while a UPS keeps the node running in the case of power failures.

2) *Social aspects of community networks:* Personal and social relationships play an important role in the community network deployment. The deployment of new nodes need the collaboration among people. If a new node is deployed, the owners of the neighbouring nodes need to connect with it, thus there has to be an interaction among the people. Two types of social networks can be observed from Guifi.net's mailing list⁵. One is at the global level of the whole Guifi.net network. In this list, technical issues are discussed. People from any part of Guifi.net community participate, and even external people who are interested can take part. The second type is the local social network, between node owners within a zone and between neighbouring zones. They use local mailing lists and some local groups also hold weekly meetings.

Guifi.net is organized into zones. A zone can be a village, a small city, a region, or a district of a larger city. The

⁵<http://guifi.net/en/forum>

organization of the group within a zone is of many types. Mostly the interests, available time and education of the people drive what happens in the zone. We note that while the allocation of IP addresses and layer 3 networking is agreed among all Guifi.net zones, as it is needed to make the IP network work, the detailed technical support is rather given within the local community of the zone. Therefore, we identify a zone to have the highest social strength within the community network.

3) *Members of community networks*: Participants of community networks are principally consumers and producers of the network. Most of them as producers contribute infrastructure and time to the networks, while as consumers they use the available services the network offers. The community network, however, is not maintained solely based on the contribution of infrastructure. Some users must also contribute with their time and knowledge. Time is needed, for instance, for maintenance tasks, which might require technical knowledge or not. Technical knowledge is required because the network is an IP network, which needs to be managed and configured.

4) *Resource sharing in community networks*: Community networks are a successful case of resource sharing among a collective. The resources shared are networking hardware but also community network participants' time that they donate, to 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. This is because even under the decentralized management of the equipment, the owner of the device ultimately has the full access and control of that network device.

Reciprocal resource sharing is, in fact, part of the membership rules or peering agreements of many community networks. The Wireless Commons License⁶ (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. Therefore, resource sharing in community networks from the equipment perspective refers in practice to the sharing of the nodes' bandwidth. This sharing, done in a reciprocal manner, enables the traffic from other nodes to be routed over the nodes of different node owners and allows community networks to successfully operate as IP networks. We observe that in most community networks the focus at the moment is on the bandwidth sharing alone. There is not much awareness about sharing other computing resources, such as storage or CPU time, inside of community networks.

5) *Ownership of nodes in community networks*: Community networks grow organically. Typically a new member that wants to connect to the community network contributes with the hardware required to connect to other nodes. A node of a community network therefore belongs to the member who is its sole owner. Such a node is normally located in the

member's premises.

Although less typical, a few nodes in Guifi.net have also been successfully crowd-funded if such a node was needed by several people. Crowd-funding of a node happened when for a group of people an infrastructure improvement was necessary. For example, an isolated zone of Guifi.net established a super node to connect to other zones. In such a case, the node has been purchased with the contributions of many people. The location of such a node follows strategic considerations, trying to optimize the positive effects on the performance that are achieved with the addition of the new infrastructure. We can see that both the options, individual ownership and crowd-funding of resources, occur in practice and could be considered for community clouds.

6) *Services in community networks*: Services and applications offered in community networks usually run on the machines that the member connects to the network and these machines are used exclusively by that member. The usage of the community network's services among its members, beyond that of access to the Internet, is however not very strong.

B. Community Cloud Scenarios

Based on the socio-technical characteristics of community networks analysed above, we start sketching our vision of community clouds.

1) *Local Community Cloud*: This scenario is derived from the topology of the community network, given by the fact that the community network generally has two different types of nodes, super nodes and ordinary nodes, and the observed characteristics of the strength of social network within zones [6]. In such a 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.

2) *Federated Community Cloud*: Multiple SNs from different zones in a community network can connect and form federated clouds [7]. SNs connect physically with other SNs through wireless links and logically in an overlay network to other SNs that manage local clouds. SNs coordinate among themselves for provisioning infrastructure service so the requests originating from one SN's zone can be satisfied by the resources allocated from another SN's zone. Figure 4 shows an example of a federated community cloud formed by SNs from three zones. The ONs in a given zone are directly managed by the SN in that zone but they can also consume resources from other zones because of the coordination among SNs.

III. ARCHITECTURE AND DESIGN

The option for enabling a community cloud in a community network on which we focus here is to deploy on SNs a cloud management system tailored to community networks. Available popular cloud management platforms, such as OpenNebula [7], can principally be applied to provide the basic management of local clouds. The architecture for the

⁶<http://guifi.net/es/ProcomunXOLN>

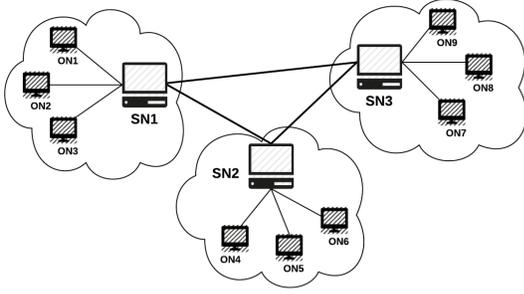


Figure 4. Super and ordinary nodes in federated community cloud

cloud management platform that we propose for community networks is shown in Figure 5 and consists of the following:

- The ONs of the community network host the virtual machine (VM) instances and constitute the hardware layer of the cloud architecture.
- The core layer residing in the SNs contains the software for managing, scheduling and monitoring the VMs running on ONs.
- The cloud coordinator is responsible for the federation of the cloud resources that are independently managed by different local community clouds.
- The front end layer provides the interface of the infrastructure service (Infrastructure-as-a-Service, IaaS).

A. Incentive Mechanism in Community Cloud

The participants in community network are mainly volunteers so it is necessary that the community cloud has incentive mechanisms in place that encourage members to contribute with their hardware, effort and time. When designing such mechanisms, one has to take into account the heterogeneity of the network, nodes and communication links, since each member brings in a widely varying set of resources and physical capacity to the system.

As detailed in [8], [9], we propose an incentive mechanism which applies reciprocity-based resource allocation. This is inspired by the Parecon economic model [10], [11] which focuses on social welfare by considering the inequality between nodes. In this model, nodes' rewards are calculated based on their effort, which is a function of their capacity as well as their contribution to the system.

The criteria that a SN uses to evaluate requests from ONs is the following: When an ON asks for a resource from a SN, which in this case is the number of VMs and the duration for which they are needed, the SN first checks whether the ON's credit is sufficient to cover the cost of the transaction. This cost is proportional to the number of VMs requested R_i and the duration T_i they are occupied.

$$transaction_cost = \gamma R_i \times \rho T_i \quad (1)$$

where γ and ρ are nonzero coefficients for the amount and duration of VMs committed respectively [9]. If the ON does not have sufficient credit, the request is rejected. However,

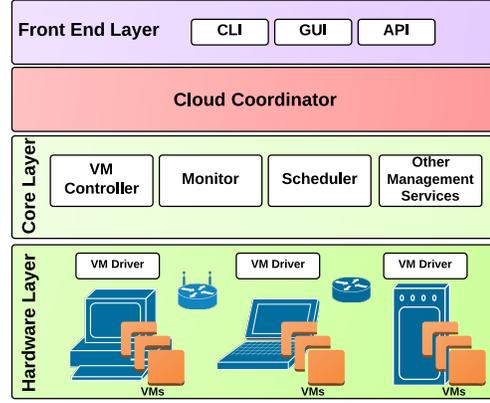


Figure 5. Architecture of community cloud management system

SN sometimes allows requests from ONs with zero or negative credit so as to encourage them to participate in the system and earn credit by contributing more VMs. If an ON has enough credits, the SN searches for VMs provided by the ONs in its zone. If the demand cannot be met locally, the SN forwards the request to other SN zones. For each ON which provides VMs, the SN calculates the transaction cost and adds it to that ON's credits, while the cost is deducted from the consumer ON's credits. Once the operation is completed, the effort for each ON involved in the transaction is recalculated as in [8] by:

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

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. The significance of this is that a node with less capacity has put in more effort than a node with higher capacity even if both of them donated same number of VMs.

B. Cloud Coordinator

Normally applications running at a local community cloud can only consume resources from the ONs directly managed by that particular SN. With the cloud coordinator, the infrastructure service can provide a unified view of the resources contributed by multiple local community clouds. Figure 6 shows the implemented components of the cloud coordinator in the prototype as described below.

- ON Management: ONs can register with SN to request and to contribute resources.
- Regulation Mechanism: When pooling resources from multiple zones, the cloud coordinator applies a regulation mechanism that takes into account resource utilization and contribution by different nodes to perform resource allocation.
- SN Interconnectivity: The design of a community cloud manager follows a decentralized approach, so cloud coordinator relies on gossip-based discovery mechanisms

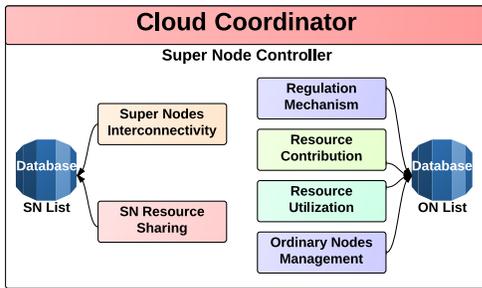


Figure 6. Components of cloud coordinator

to manage overlay network of the SNs in community cloud. The updated list of adjacent SNs is saved in SN-List database.

- SN Resource Sharing: When requests from ONs cannot be met from locally available resources, SN can request resources from other SNs in the system.

C. Interaction between Super and Ordinary Nodes

The overlay network that results from the hierarchical architecture of the community cloud is formed by SNs. The difference between SNs and ONs, from the point of view of cloud management, is that SNs support greater functionality for handling VMs. A SN has full installation of the cloud management software and so enables the user to manage VMs executing on ONs. In most cases, a SN will be a comparatively stable node, most likely connected to a hub of the wireless mesh network. Each SN is responsible for a set of ONs and manages their metadata in its ON-List database. SNs publish their status and details of local resources to other SNs, e.g. by gossiping, and each SN stores this information in SN-List database. ONs, on the other hand, only act as hosts for executing VMs. Most components of cloud management software are not installed on ONs, so the VMs cannot be controlled conveniently from the ONs themselves. Each ON registers to a parent SN by providing it with the list of the available resources, and the parent SN is responsible for management of VMs. ONs periodically send a heartbeat message to their parent SN to inform it about their current status.

IV. PROTOTYPE IMPLEMENTATION

We have implemented a prototype of the incentive-based regulation mechanism that was proposed in [8], [9]. We implemented the components in the Python programming language and used CouchDB⁷ as database. We chose Python because the current host operating system installed on ONs is OpenWRT⁸, which supports Python, but does not support many other languages such as Java. We selected CouchDB because among its advantages, it is lock-free, schema-less and provides a REST interface, and is also part of other

⁷<http://couchdb.apache.org>

⁸<http://openwrt.org>

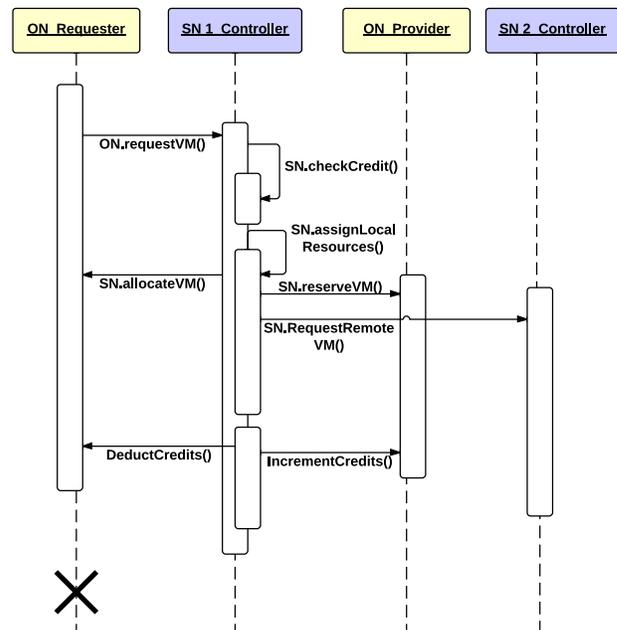


Figure 7. Details of the VM request operation by an ON

components of the SN's cloud management software being developed. In the SNs, Debian operating system is installed.

ONs use the remote procedure call (RPC) mechanism to connect to the SN. First of all, an ON assigns itself to a parent SN with a register message which includes metadata of that ON such as IP address, total capacity and number of VMs shared. This registration information is stored in the ON-List database of the parent SN by creating an entry for the corresponding ON. After that, the ON is ready to send request messages to its parent SN. Figure 7 shows the request processing algorithm followed by SN [9]. When an ON requests its parent SN for any VMs, it specifies the duration for how long it needs to use the VMs. This request is evaluated by performing incentive and decision mechanisms as explained in section III. If a request cannot be met locally, the corresponding parent SN checks its SN-List database to find another zone with available resources. The interactions between SNs are also made through RPC mechanism. In the SN controller software, there is a separate process which regularly checks the database for any updates. If the duration of a consumer ON's resource request has expired, it frees the VMs and makes them available again for the provider ON, and updates the metadata entries of the corresponding ONs in the ON-List database. The current implementation keeps track of the number of VMs contributed and consumed by each ON. The system copes with ONs connecting and disconnecting from the SN at any time since ONs periodically send heartbeat messages to the SN. The design allows us to include, in addition, values of metrics like CPU, memory and bandwidth usage, which in the future could be used for fine-grained decisions on resource assignments.

Table I
RESOURCE CAPACITY OF NODES

Node IDs	Capacity	Shared VMs
f101	2	1
f102	3	3
f103	3	1
f104	1	1

V. EVALUATION

We deploy the prototype of the regulation component of the cloud coordinator from community cloud manager in the Community-Lab⁹ testbed [12], which is provided by the CON-FINE European project¹⁰ [13]. The cloud coordinator components are installed on nodes of the Community-Lab testbed, which consist of devices of the model Jetway JBC372F36W, as introduced in Figure 3. Depending on the experiment, one or two nodes operate as SNs, while each ON hosts between one and four VM instances. The objectives of the experiments are twofold:

- 1) Experiment 1: Assess the prototype operation regarding the incentive-based resource assignment algorithm in a local community cloud scenario.
- 2) Experiment 2: Study the coordination between SNs from different zones in the federated community cloud scenario with heterogeneous resource distribution.

A. Experiment 1: Resource Assignment in Local Community Cloud Scenario

In order to study the performance of the prototype in a real deployment of a local community cloud, we install our software components in five nodes of Community-Lab testbed, which are connected to the Guifi.net community network. Each node behaves as an ON but with different configuration, in order to have a heterogeneous set of cloud resources. Table I shows the capacity of the different nodes in terms of VM instances and the number of VMs made available for sharing with other nodes. For instance, node f102 shares all of its available capacity, while node f103 shares only one-third of its capacity with the other nodes. In this experiment, one node acts as SN while four nodes act as ONs which connect to the single SN. Each ON sends request for VM instances to the SN at regular intervals. VMs are requested for 20 seconds interval at a time. Each ON requests as many VMs as its total capacity, for example node f101 always requests 2 VMs. If the request is accepted by the SN, the ON obtains the VMs for the next 20 seconds. If the request is rejected, the ON waits for 5 seconds before making any further requests. The experiment is run with this setup for around 5 minutes. We analyse the different aspects of the system behaviour in the following.

1) *Resource Utilization*: Figure 8 shows the level of resource utilization in the system in terms of the number of reserved VMs versus the total number of VMs. It can be seen

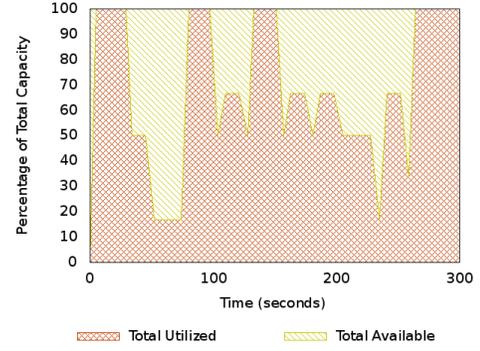


Figure 8. Overall resource utilization of the four ONs

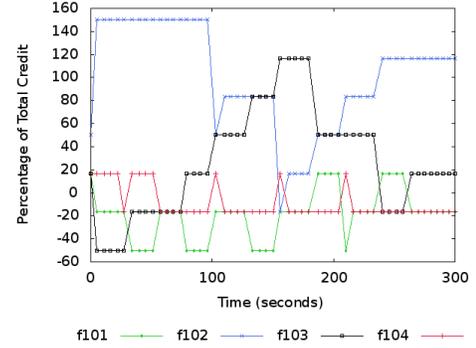


Figure 9. Distribution of credit among the four ONs

that resource utilization varies widely and 100% utilization, meaning all the VMs are occupied, occurs only for short intervals. This is because as nodes obtain VMs, they spend their credit and can no longer request further VMs. At approximately second 80, the utilization gets very low. Nodes then need to earn credits by providing VMs to others before they can request VMs again. So even though VMs are available, they cannot be utilized due to the lack of credit in the system.

2) *Credit Distribution*: Figure 9 shows the credit distribution among the four ONs during the 5 minutes of the experiment. A node's credit is affected by how many VMs it shares and how much credit it spends to obtain VMs. When a node shares most of its capacity, like ON f102 providing all its 3 VMs, it earns more credits and so maintains a high credit level during the experiment. On the other hand, when a node continuously consumes VMs like ON f101 and f104, it keeps on spending its credit which does not go beyond a certain level. Of particular interest is the behaviour of ON f103, which earns credit in the start and gets a spike in credit level halfway through the experiment, but then quickly spends it as it requests VMs from others. Note that an ON's credit can be negative or higher than 100% of the total credit because in the current implementation SN can allow requests from ONs with zero or negative credit.

3) *Success Ratio*: Figure 10 shows the ratio of the fulfilled requests for each node, which is affected by the level of credit of the node and the amount of resources available in the

⁹<http://community-lab.net>

¹⁰<http://confine-project.eu>

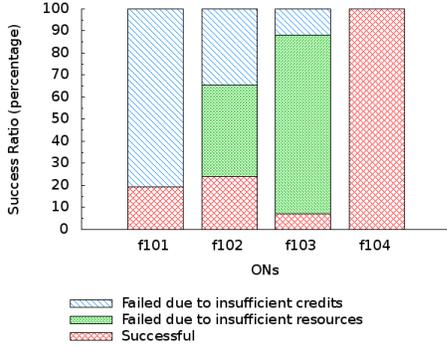


Figure 10. Ratio of fulfilled and rejected requests for each ON

system. ON f104 has the most success since it requests only one VM at a time while ON f103 has the least success since it requests 3 VMs, which is half of the total shared VMs in the system. ON f101, on the other hand, gets requests rejected because of the lack of credit. Therefore, this node has to wait to gain the needed credits.

B. Experiment 2: Resource Assignment in Federated Community Cloud Scenario

In this experiment, we set up two local clouds, each with one SN and four ONs to study the federated community cloud scenario, as illustrated in Figure 4. Table II shows the two cloud cases with different number of VMs available in the two zones. In the case of scarce capacity (case 1), the nodes in the SN1 zone share very few VMs compared to nodes in SN2 zone. In the case of equal capacity (case 2), the nodes in both the zones share the same number of VMs.

Figure 11 shows the proportion of the requests fulfilled by VMs provided by the other zone. With scarce capacity in SN1 zone, around 50% of the requests are fulfilled by VMs provided by SN2 zone. SN2 with sufficient capacity is able to meet most of the requests from VMs within the same zone, forwarding less than 15% requests to the other zone. In the second case, when both zones have the same available capacity, most of the requests get processed within the same zone for both the SNs. This shows that a federated community cloud scenario extends the resources assigned to zones with limited capacity.

C. Discussion

From the results of these experiments, we observed that:

- 1) The prototype of the regulation service deployed in real community network nodes performed the required operations. Its components worked correctly both in the ON's host operating system (OpenWRT) and the SN's operating system (Debian). We could not observe the limitations of our implementation within scales that are realistic for community networks. We note however that as a continuation of this work a more extensive deployment of several federated community clouds with real users and real usage should ultimately be undertaken.

Table II
TWO CASES WITH DIFFERENT RESOURCE DISTRIBUTION BETWEEN ZONES

SNs	ONs	Case 1: Scarce Capacity		Case 2: Equal Capacity	
		Total VMs	Shared VMs	Total VMs	Shared VMs
SN1	ON1	3	1	3	2
	ON2	3	1	3	3
	ON3	3	1	3	2
	ON4	1	1	1	1
SN2	ON1	3	2	3	2
	ON2	3	3	3	3
	ON3	3	2	3	2
	ON4	1	1	1	1

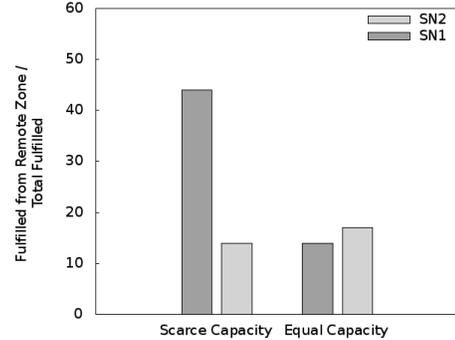


Figure 11. Resource assigned over different SN zones

- 2) The algorithm used for the regulated resource allocation controlled the VM assignments, taking into account the user's contribution and usage. More complex situations, however, should be created in further studies to provide additional insight into how the system behaves.
- 3) Our experiments were carried out on limited number of nodes and for limited time. If our prototype was deployed on additional nodes that are geographically widely spread and run for extended periods, the VM assignment decisions might need to take into account information from network awareness, to select the appropriate cloud resource providers.

VI. RELATED WORK

There are only a few research proposals for community cloud computing [5]. Most of them do not go beyond the level of an architecture, and at most a practical implementation is presented. None of these implementations, to our knowledge, are actually being deployed inside of real community networks.

The Cloud@Home [14] project aims to harvest in resources from the community for meeting the peaks in demand, working with public, private and hybrid clouds to form cloud federations. The authors propose a rewards and credit system for ensuring quality of service. Social cloud computing [15] takes advantage of the trust relationships between members of social networks to motivate contribution towards a cloud storage service. Users trade their excess capacity to earn virtual

currency and credits that they can utilize later, and consumers submit feedback about the providers after each transaction which is used to maintain reputation of each user.

Various incentive mechanisms have been studied in the literature in the context of peer-to-peer systems [16] that address different problems of distributed resource sharing. Distributed voluntary computing platforms, like Folding@home [17] and Seattle [18], on the other hand, mainly rely on altruistic contribution of resources from the users.

From the review of related work, we find that none of the above cases correspond to the concrete situation of community networks such as targeted by us. In the cloud system that we propose, we aim to take into account several of the important factors that characterize community networks, such as the scenarios we identified from the conditions of community networks, and the cloud architecture we considered is tailored to these scenarios. We also put emphasis on implementing the proposed components as prototype and test them in deployments on real nodes to identify practical issues. We note that compared with the related work, we follow the approach of developing a prototype that should contribute to building a production community cloud system to be used in real community networks.

VII. CONCLUSION AND FUTURE WORK

Community clouds are motivated by the additional value they would bring to community networks. Applications and services deployed upon community clouds would boost the usage and spread of the community network model as ICT infrastructure for society. As such, it is timely to research on clouds for community networks, since mainstream cloud computing technologies are mature now and are widely used in today's Internet.

The paper analyses the key socio-technical characteristics of community networks in order to derive two community cloud scenarios, the local community cloud and the federated community cloud. These scenarios are targeted by a community cloud architecture which is proposed, with the need for an incentive-driven regulation mechanism identified as a key component to encourage contribution towards and foster adoption of community clouds. The regulation component was implemented as a prototype, and evaluated in an experimental deployment on real community network nodes to explore its behaviour for both community cloud scenarios.

Carrying onwards from the experience and results with this prototype, a working service needs to be developed further that provides the feedback loop between the users' contribution and experience, which will be inevitable for adoption, sustainability, maintenance and growth of cloud infrastructures in community networks. Larger scale deployments are required with extended implementation of the different components of the community cloud architecture. This should be complemented by additional services and applications deployed in the cloud infrastructure, which will provide enhanced value and utility to the members of community networks for their contribution towards the community cloud.

ACKNOWLEDGEMENT

This work was supported by the European Framework Programme 7 FIRE Initiative projects CONFINE, FP7-288535, and CLOMMUNITY, FP7-317879, and by the Universitat Politècnica de Catalunya BarcelonaTech and the Spanish Government through the Delfin project, TIN2010-20140-C03-01.

REFERENCES

- [1] "Guifi.net new sections of fiber deployed to the farm." 2012. [Online]. Available: <http://en.wikinoticia.com/Technology/internet/122595>
- [2] A. M. Khan, L. Sharifi, L. Veiga, and L. Navarro, "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, Oct. 2013.
- [3] J. Jiménez, R. Baig, P. Escrich, A. M. Khan, F. Freitag, L. Navarro, E. Pietrosevoli, M. Zennaro, A. H. Payberah, and V. Vlassov, "Supporting Cloud Deployment in the Guifi.net Community Network," in *5th Global Information Infrastructure and Networking Symposium (GIIS 2013)*, Trento, Italy, Oct. 2013.
- [4] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *NIST Special Publication*, vol. 800, no. 145, 2011.
- [5] A. Marinos and G. Briscoe, "Community Cloud Computing," *Cloud Computing*, vol. 5931, pp. 472–484, 2009.
- [6] D. Vega, L. Cerda-Alabern, L. Navarro, and R. Meseguer, "Topology patterns of a community network: Guifi.net," in *1st International Workshop on Community Networks and Bottom-up-Broadband (CNBuB 2012)*, within *IEEE WiMob*, Barcelona, Spain, Oct. 2012, pp. 612–619.
- [7] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "IaaS Cloud Architecture: From Virtualized Datacenters to Federated Cloud Infrastructures," *Computer*, vol. 45, no. 12, pp. 65–72, Dec. 2012.
- [8] U. C. Büyüksahin, A. M. Khan, and F. Freitag, "Support Service for Reciprocal Computational Resource Sharing in Wireless Community Networks," in *5th International Workshop on Hot Topics in Mesh Networking (HotMESH 2013)*, within *IEEE WoWMoM*, Madrid, Spain, Jun. 2013.
- [9] A. M. Khan, U. C. Büyüksahin, and F. Freitag, "Towards Incentive-Based Resource Assignment and Regulation in Clouds for Community Networks," in *Economics of Grids, Clouds, Systems, and Services*, ser. Lecture Notes in Computer Science. Zaragoza, Spain: Springer International Publishing, Sep. 2013, vol. 8193, pp. 197–211.
- [10] R. Rahman, M. Meulpolder, D. Hales, J. Pouwelse, D. Epema, and H. Sips, "Improving Efficiency and Fairness in P2P Systems with Effort-Based Incentives," in *2010 IEEE International Conference on Communications (ICC 2010)*, Cape Town, South Africa, May 2010.
- [11] D. Vega, R. Meseguer, S. F. Ochoa, and F. Freitag, "Sharing hardware resources in heterogeneous computer-supported collaboration scenarios," *Integrated Computer-Aided Engineering*, vol. 20, no. 1, pp. 59–77, 2013.
- [12] A. Neumann, I. Vilata, X. Leon, P. Escrich, L. Navarro, and E. Lopez, "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*, Oct. 2012.
- [13] B. Braem, R. Baig Viñas, A. L. Kaplan, A. Neumann *et al.*, "A case for research with and on community networks," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 3, pp. 68–73, Jul. 2013.
- [14] S. Distefano and A. Puliafito, "Cloud@Home: Toward a Volunteer Cloud," *IT Professional*, vol. 14, no. 1, pp. 27–31, Jan. 2012.
- [15] M. Puceva, I. Rodero, M. Parashar, O. F. Rana, and I. Petri, "Incentivising resource sharing in social clouds," *Concurrency and Computation: Practice and Experience*, Mar. 2013.
- [16] X. Shen, H. Yu, J. Buford, and M. Akon, *Handbook of Peer-to-Peer Networking*. Springer Heidelberg, 2010, vol. 1.
- [17] A. L. Beberg, D. L. Ensign, G. Jayachandran, S. Khaliq, and V. S. Pande, "Folding@home: Lessons from eight years of volunteer distributed computing," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, Rome, Italy, May 2009.
- [18] J. Cappos, I. Beschastnikh, A. Krishnamurthy, and T. Anderson, "Seattle: a platform for educational cloud computing," in *40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)*, Chattanooga, USA, Mar. 2009, pp. 111–115.