Towards Incentive-Compatible Pricing for Bandwidth Reservation in Community Network Clouds

Amin M. Khan[1,2], Xavier Vilaça[2], Luís Rodrigues[2], and Felix Freitag[1]

[1] Department of Computer Architecture
Universitat Politècnica de Catalunya. Barcelona, Spain
{mkhan, felix}@ac.upc.edu
[2] Instituto Superior Técnico, Universidade de Lisboa
INESC-ID Lisboa. Lisbon, Portugal
{xvilaca, ler}@tecnico.ulisboa.pt

Abstract. Community network clouds provide for applications of local interest deployed within community networks through collaborative efforts to provision cloud infrastructures. They complement the traditional large-scale public cloud providers similar to the model of decentralised edge clouds by bringing both content and computation closer to the users at the edges of the network. Services and applications within community network clouds require connectivity to the Internet and to the resources external to the community network, and here the current best-effort model of volunteers contributing gateway access in the community networks falls short. We model the problem of reserving the bandwidth at such gateways for guaranteeing quality-of-service for the cloud applications, and evaluate different pricing mechanisms for their suitability in ensuring maximal social welfare and eliciting truthful requests from the users. We find second-price auction based mechanisms, including Vickrey and generalised second price auctions, suitable for the bandwidth allocation problem at the gateways in the community networks.

Keywords: community clouds; community networks; auctions; resource allocation

1 Introduction

Community network clouds represent efforts to collaboratively build cloud infrastructures in the community networks [1,2], extending the ideas from the volunteer computing model to community cloud computing [12]. Community networks are a successful example of social collective for building ICT infrastructure for the communities in a bottom-up fashion, for instance Guifi.net, currently the largest community network in the world, connects more than 28,000 locations (nodes) with wireless and optical fibre links [1]. Other examples of successful community networks include Athens Wireless Metropolitan Network (AWMN) in Greece, Freifunk in Germany, FunkFeuer in Austria, and Ninux in Italy.

Community network clouds build on this success of community networks and aim to provide services and applications of local interest for the communities by applying the model cloud computing. They fit nicely with the recent shift in exploring alternative approaches to large-scale data centres based public cloud computing, which include Inter-Cloud and federated clouds (where multiple public cloud providers work together), hybrid clouds (where enterprises combine their own cloud infrastructure with the public clouds), multi-clouds (where applications procure services from different public cloud providers), community clouds (where dedicated infrastructure is provided for a specific community), and edge clouds using nano data centres [14] (where smaller clusters are deployed at the edges of the network to avoid latency and improve content-delivery). These initiatives provide an excellent backdrop to explore the role of the community network clouds in enhancing the value proposition of the community networks, since an infrastructure of nano data centres [14] to be deployed in a community network has to fit well with specific socio-economic and technical context of the community networks [10]. Figure 1 shows how such an edge cloud can be deployed within a community network. The servers are present at different locations, either caching content for media-rich applications or performing computation locally for time-critical applications, and which require connection to the data centres through the Internet, for which they rely on the gateway providers available in the community network [1].

When cloud applications are deployed within community networks, in many cases connectivity external to the community network is important. In the basic case, cloud applications may want to backup or synchronise data with servers external to the community network, or require fetching data for operating the service, for instance a video-on-demand service may download fresh content. Also, a service available in multiple community networks requires access at the gateways for exchanging data, and gateways in this case act to federate the community networks. Applications from Internet of Things and smart cities involve collecting data from the sensors, which may have to be shared with servers outside the community network for data analysis. For the case of edge clouds, the servers residing within the community networks, acting as nano data centres, require connectivity to the data centres. In all these situations, the applications deployed on servers within the community network require bandwidth at the gateways with quality-of-service (QoS) guarantees to connect to the Internet, though their requirements for prioritising, robustness, waiting time, and throughput, may vary for different scenarios.

Various mechanisms have been extensively studied in the literature for pricing bandwidth in wireless networks, including application of game theory and auctions [11]. These concepts include static pricing approaches, like fixed usage-based pricing where all users are charged the same amount per unit of bandwidth, and priority pricing where users pay differently according to the priority class of the requests, and dynamic market-based pricing mechanisms, which can be based on auctions like sealed first-price, generalised second price (GSP) [7], and Vickrey–Clarke–Groves (VCG) auctions [5]. In this paper, we study these pricing

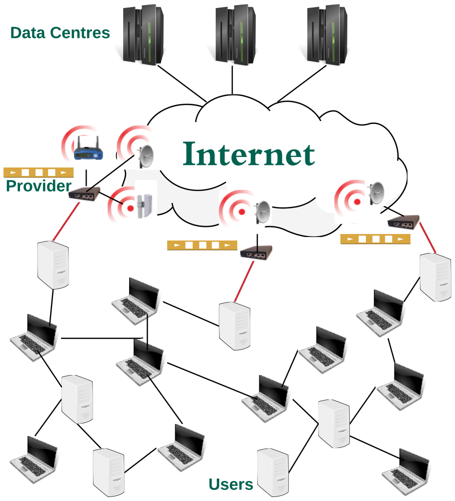

Fig. 1. Users connected to the service provider's gateway in a community network

mechanisms with the goals of maximising social welfare (maximum utility for maximum number of users), and truthfulness (users declare their true valuation to the provider). Of the above mechanisms, only VCG ensures maximum social welfare as well as truthfulness [13], as long as the optimal allocation of resources can be computed in polynomial time. In practice, however, many resource allocation problems involve combinatorial optimisation and are NP-Hard, so often approximation [19] or heuristics [5] based approaches are employed which guarantee truthfulness but not maximal social welfare.

Recent work has explored auctions for incentivizing bandwidth sharing in community networks [18, 21], and allocating bandwidth in public clouds [5, 15, 20] and grid systems [4]. However, the problem of bandwidth allocation for applications in community network clouds has largely been unexplored to the best of our knowledge. In community network clouds, there are multiple independent bandwidth providers with no centralised control, and, moreover, the users are connected to the providers through a multi-hop community network (in contrast to the dedicated networks within large-scale data centres). Community networks are a social collective [16], where most resources are contributed on a volunteer and reciprocal basis. The gateways provide access to the Internet on a best-effort basis, so providing guaranteed bandwidth to cloud applications is critical to the successful operation of cloud applications deployed in community network clouds.

Our contribution in this paper is showing the applicability of decentralised edge clouds model [14] to the community network clouds, and framing the problem of bandwidth reservation at the gateways as crucial for cloud applications to function and flourish in community network clouds. We provide a model that differentiates between cloud applications with different priority classes, and use this model to evaluate the suitability of different pricing mechanisms in the literature to the problem of bandwidth reservation in community network clouds.

The rest of the paper is organised as follows. We relate our work to the state-of-the-art in section 2. We present our model in section 3 for analysing different pricing mechanisms for reserving bandwidth, and in section 4, we evaluate them through simulation experiments. We conclude and indicate future work in section 5.

2 Related Work

Community clouds built using resources contributed by the community have garnered interest recently [12], with most work focusing on the exchange of virtualised resources. Along with incentives-based resource regulation for providing and consuming virtual machines in community network clouds [8, 9], other approaches have focused on social cloud computing [3] to share storage and computation resources among the users of online social networks.

In respect of bandwidth resources, community networks require incentives as communication relies on cooperation among the users. The recent literature has modelled this in non-cooperative game theory [18, 21]. Community networks solve this problem to a large extent through social mechanisms like enforcing reciprocal sharing agreements [1]. Bandwidth reservation has also been under focus recently in the grid systems [4], and in public clouds, both for internal bandwidth within data centre networks [6], and external bandwidth [5, 15, 20] to the Internet.

Our work differs since we assume the co-operation among users because of the social institution of community networks, so traffic from other nodes is guaranteed to transit on the intermediary nodes. This also follows from the fact that the bandwidth available within community networks is not priced, and is symmetric, i.e. upstream capacity is same as downstream, and under normal use sharing does not incur costs for the node owners [1]. Non-cooperative users are either excluded from the system, or correct their behaviour because they are penalised by the community. We focus on the cloud applications that are deployed in community network cloud, which can be bandwidth-intensive generating significant traffic flows within the community network, but also require guaranteed and stable connection to the Internet through the gateways in the community network.

3 System Model

We consider a bandwidth provider $\mathbb{P}$ in the community network and a set of N users $\{1, 2, \dots N\}$. The provider operates a gateway to the Internet, allowing access external to the community network to the users. The users are connected to the provider's gateway through the wireless and fibre links in the community network [1], and the applications in community network cloud access their external servers and Internet through this gateway. Figure 1 shows the users in the community network connected to the provider through multiple such paths, where only few of these users are the clients of the provider for reserving bandwidth.

The provider processes the requests in a queue at the gateway, where time in the queue is divided into an infinite sequence of slots starting from 1, where all the available bandwidth is allocated to exactly one user in each slot. The provider allocates the slots in batch after receiving all the requests from N users and assigns the next N slots, one to each user.

For the sake of simplicity, we divide the users into two priority classes, $h \in \{0,1\}$, some have lower priority requests, h_0, and some have higher priority requests, h_1. Here in this model, the main consideration for higher priority requests is that they are more sensitive to the waiting time, and prefer to reserve earlier slots in the queue. Provider $\mathbb{P}$ aims for an optimal schedule when allocating the slots to the users, so as to maximize its revenue and the overall utility for all the users. We provide formal details below.

Schedule: A schedule ϕ maps each time slot t to a user i.

Value: For any schedule ϕ and user i, let t be the slot assigned to user i, then $v_i(h,t)$ is the valuation given by i being allocated for time slot t, where $h \in \{0,1\}$ is the priority class of the user. v_i is communicated by each i to $\mathbb{P}$ beforehand.

Utility: For any schedule ϕ which assigns user i a slot t, the utility $u_i(\phi,t)$ for user i is difference between the value v_i and the payment $p_i(\phi,t)$ made by user i to $\mathbb{P}$.

$$u_i(\phi,t) = v_i(h,t) - p_i(h,t) \tag{1}$$

Restriction: Any slot t can be assigned to at most one user.

Optimization: Find ϕ that maximizes the social welfare, which is the sum of utilities u_i of all users, while fulfilling the restrictions.

$$\text{maximise} \quad welfare(\phi) = \sum_{i \in N} u_i(\phi,t) \tag{2}$$

Scheduler: Function S that maps $\boldsymbol{u} = (u_i)_{i \in N}$ to optimal ϕ.

Goal: A user i when submitting the request to $\mathbb{P}$, declares the priority class h_i and value v_i, and also the bid amount b_i where applicable. When the user behaves truthfully the reported value v_i^* is the same as her inherent value v_i. We want to ensure that it is truthful for every i to declare her true value of v_i, regardless of the declared values v_j for any $j \neq i$. Such a mechanism is said to be truthful in dominant strategy, where users have no incentive to misreport their values [13].

3.1 Pricing Mechanisms

Given the above model, the prices are calculated for the bandwidth usage according to different mechanisms [11].

Fixed Pricing In the case of fixed usage-based pricing, all the users pay the identical price c_0 for each unit of bandwidth consumed, which is constant irrespective of the priority class.

Priority Pricing In the case of priority pricing, users pay according to the priority class h. Since in our model, there are only two priority classes h_0 and h_1, provider $\mathbb{P}$ charges two different prices c_{h_0} and c_{h_1} per unit of bandwidth, respectively.

First-Price Auction In the case of sealed first-price auction, users make different bids b_i depending on their priority class h, with high priority requests quoting higher bid amounts in general. Each winning user pays their bid amount.

$$p_i(\phi, t) = b_i \tag{3}$$

Generalised Second Price (GSP) Auction In a generalised second price (GSP) auction [7], users make different bids b_i but in this case the winning user pays the amount corresponding to the next highest bidder. So the user with the highest bid, pays the amount of the second highest bidder, the second highest bidder pays the amount of the third highest bidder and so on.

Vickrey-Clarke-Groves (VCG) Auction VCG is a second-price sealed-bid auction based mechanism, which ensures truthfulness and maximum social welfare [13], if the provider $\mathbb{P}$ can calculate optimal schedule ϕ in polynomial time. Each user i provides a bid b_i to $\mathbb{P}$, and given a schedule ϕ, each user i pays the price $p_i(\phi, t)$ according to:

$$p_i(\phi, t) = \sum_{\substack{j!=i \\ j \in N}} (v_j(h, \phi') - b_j) - \sum_{\substack{j!=i \\ j \in N}} (v_j(h, \phi) - b_j) \tag{4}$$

where ϕ and ϕ' are the schedules that maximise $\sum_{i \in N} u_i$ while including and excluding the bid b_i by user i from the allocation respectively.

3.2 Scheduling Algorithm

We consider a simple scheduling algorithm which applies a greedy approach for mapping users' requests to the available slots. Algorithm 1 shows the scheduling algorithm, where $\mathbb{P}$ assigns the slots to the users in non-increasing order of their reported bids (and corresponding h_i and v_i) for the bandwidth resource. The prices calculated are dependent on the pricing mechanism, the priority class h of the requests, and the assigned slot t in the schedule ϕ. The runtime of the algorithm is $\mathcal{O}(N \, log \, N)$ for N users, however, VCG mechanism requires computing N schedules for calculating payments for the N winning bids, so the running time in the case of VCG is $\mathcal{O}(N^2 \, log \, N)$.

The greedy approach, in general, does not always provide an optimal allocation, which is a pre-requisite for VCG mechanism. However, in the case of the model given above and considering the step function we are going to use for $v_i(h, t)$ from Figure 2, the greedy approach from Algorithm 1 always returns an optimal allocation. This can be proven through induction, and can be explained intuitively as choosing the requests with higher bids first (corresponding to higher h_i and v_i) always gives the maximum social welfare, since the value function

Algorithm 1 Scheduling algorithm for ϕ, allocating $\overrightarrow{\boldsymbol{t}}$ slots to N users

```
Input: List of users $\overrightarrow{\boldsymbol{n}}$, bids $\overrightarrow{\boldsymbol{b}}$, for total $N$ users
Output: List of assigned slots $\overrightarrow{\boldsymbol{t}}$, and payments $\overrightarrow{\boldsymbol{p}}$
Sort $\overrightarrow{\boldsymbol{n}}$ users in non-increasing order on their bids $\overrightarrow{\boldsymbol{b}}$
for $i = 1, \ldots N$ do
    $\overrightarrow{\boldsymbol{t}}[i] \leftarrow i$                                  ▷ Assign slots
end for
for $i = 1, \ldots N$ do
    $\overrightarrow{\boldsymbol{p}}[i] \leftarrow payment(\overrightarrow{\boldsymbol{b}}[i], \overrightarrow{\boldsymbol{t}}[i])$        ▷ Calculate payments
end for
```

in Figure 2 is non-increasing with time and choosing a bid with lower amount causes a loss in social welfare which can't be recovered as the time progresses.

4 Performance Evaluation

We conduct the simulation experiments using the multi-agent programmable modelling environment NetLogo [17]. In all the experiments, we consider a single provider and 500 users. We run the experiments for 1000 rounds, and plot the average values in the graphs.

For different pricing mechanisms (as explained in § 3.1), we use the following values. For fixed pricing, we set $c_0 = 0.5$. For priority pricing, we set $c_{h_0} = 0.25$ and $c_{h_1} = 0.75$. For auctions based pricing, the bids for lower priority requests h_0 are uniformly distributed in the range $[0.25, 0.5]$, while the bids for higher priority requests h_1 are uniformly distributed in the range $(0.5, 0.75]$. For differentiating between the two priority classes, we choose different time-utility functions (TUF), which in this case we have chosen as step functions for simplicity. According to this step function, the value $v_i(h,t)$, based on priority class h and slot t in schedule ϕ, decreases for both higher and lower priority classes after a threshold $t_0 = \frac{N}{2}$, as shown in Figure 2. Specifically, for lower priority class h_0:

$$v_i(h_0, t) = \begin{cases} 1.5 & \text{if } t \leq \frac{N}{2} \\ 1 & \text{if } \frac{N}{2} < t \leq N \end{cases} \tag{5}$$

And for high priority class h_1:

$$v_i(h_1, t) = \begin{cases} 3 & \text{if } t \leq \frac{N}{2} \\ 2 & \text{if } \frac{N}{2} < t \leq N \end{cases} \tag{6}$$

Each user i submits exactly one request to $\mathbb{P}$, declaring her priority class h_i, value v_i, and bid amount b_i where applicable. Both the priority classes, h_0 and h_1 occur with the same probability, so almost half of the requests are of higher priority, and the rest are of lower priority. We model lying behaviour of the users, by randomly flipping their reported priority class h to $\mathbb{P}$, according to a uniform

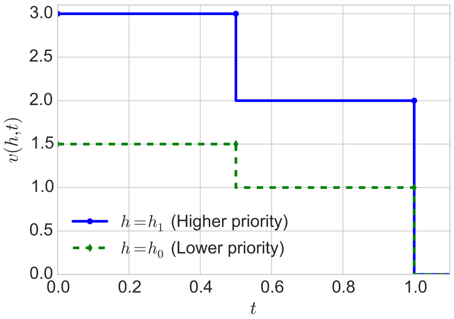

Fig. 2. Value function $v_i(h, t)$ for user i based on priority class h and slot t in schedule

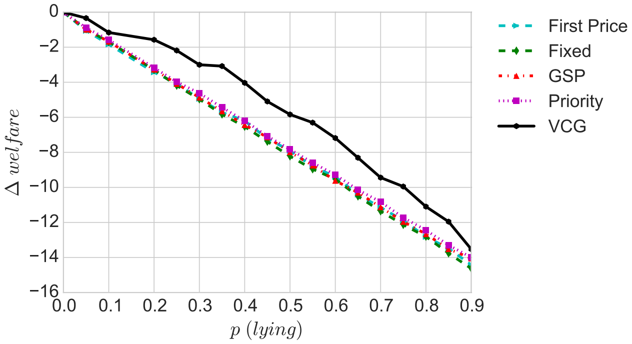

Fig. 3. Percentage difference in social welfare as more users lie

distribution. When the users lie, we observe the normalized difference from the case where all the users are truthful. Here, $u_i^*(\phi, t)$ indicates the case where the users lie to $\mathbb{P}$, and $u_i(\phi, t)$ where all the users are truthful.

$$\Delta\ welfare = \frac{\sum\limits_{i \in N} u_i^*(\phi, t) - \sum\limits_{i \in N} u_i(\phi, t)}{\sum\limits_{i \in N} u_i(\phi, t)} \tag{7}$$

Figure 3 shows how social welfare is affected when the probability of a user misreporting her value to $\mathbb{P}$ – $p(lying)$ – increases up to the point where 90% of the users may be lying. As expected, social welfare decreases as the probability of lying increases, since $\mathbb{P}$ fails to allocate better slots for higher priority requests. All the pricing schemes behave similarly as the proportion of lying users increases, except VCG which performs marginally better in that social welfare is slightly higher for VCG as compared to the other schemes. This shows the importance of encouraging truthful behaviour in the users for maximising social welfare.

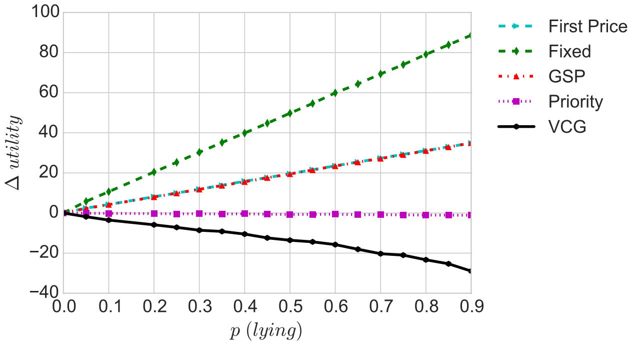

Fig. 4. Percentage difference in utility for low priority class h_0

To understand how the pricing mechanisms incentivize truthfulness for different priority classes, in the next experiment we look at the normalised difference in utility for an individual user (on average), separately for h_0 and h_1. Here again, $u_i^*(\phi, t)$ is the individual utility when some of the users lie, and $u_i(\phi, t)$ is when all the users are truthful.

$$\varDelta\ utility = \frac{u_i^*(\phi, t) - u_i(\phi, t)}{u_i(\phi, t)} \tag{8}$$

Figure 4 shows the percentage difference in the average utility for all the users with low priority requests. Note that this average is over all the users in h_0, and not only those who lie. Users from h_0 may lie in order to get higher value (through reserving an earlier slot), hoping to still pay as less as possible. Figure 4 shows that for fixed usage-based price, they do gain in utility since they are paying the same amount for a better service. For priority pricing, they gain nothing as any gains in utility are offset by the higher price. For first price and GSP auctions, the results are similar and there are gains due to lying, though less than those in the case of the fixed price. The first price and GSP auctions behave similarly since expected payments are same in the first and second price auctions, when the bids are independent and identically distributed [11], as is the case in this experiment. VCG performs better since the utility decreases when the users lie.

Figure 5 shows the percentage difference in the average utility for all the users with high priority requests. Note that this average is over all the users in h_1, and not only those who lie. Users from h_1 may lie in order to save on their payments, with the hope that they can still get the same value (through keeping their earlier slot). Figure 4 shows that users from h_1, in general, lose by lying since there is little chance that $\mathbb{P}$ will assign earlier slots to the users declaring low priority to $\mathbb{P}$. So even though they save on the payments, the decrease in value because of getting assigned later slots results in net loss for users from h_1.

In the next experiment, we look specifically at the utility for the users that report untruthful values to $\mathbb{P}$, to see the maximum gain they can get in the utility

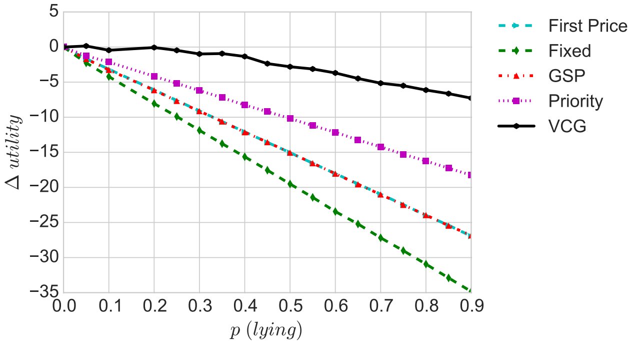

Fig. 5. Percentage difference in utility for high priority class h_1

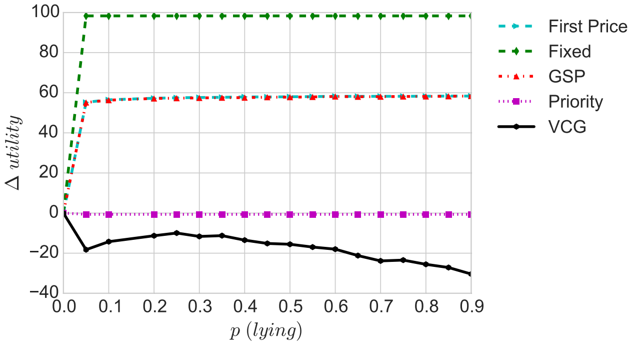

Fig. 6. Maximum gain in utility for a user from low priority class h_0

under different pricing mechanisms. Figure 6 shows the maximum gain in utility a user from h_0 can get as the number of lying users increases. Note that in this case we pick only the maximum utility for a user from h_0 that is lying, averaged across all the experiment runs. The results are similar to what we observed earlier in Figure 4.

Similarly, Figure 7 shows the maximum gain in utility a user from h_1 can obtain through lying. We noticed in Figure 5 that on average the users from h_1 do not gain through lying, but here we see that for all the pricing mechanisms except VCG, the utility for a lying user with high priority request increases with increase in the number of lying users, though the net gain is not significant. For VCG, the number of lying users does not have much impact, and the loss in utility for the lying user remains almost the same. Moreover, first price and GSP auctions perform better than priority pricing here. The user can have a net gain in utility by misreporting her priority class as h_0 when more than half of the users are lying in the case of priority pricing.

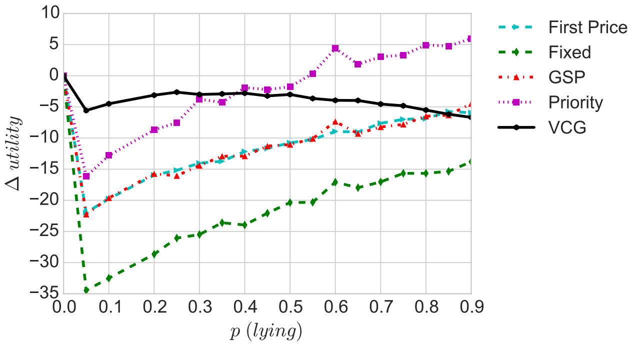

Fig. 7. Maximum gain in utility for a user from high priority class h_1

4.1 Discussion

We find that static pricing schemes like fixed usage-based pricing and priority pricing are not very useful for arbitration between requests from different priority classes, since it is hard to avoid everyone reporting their requests as higher priority [11]. Dynamic pricing, for example, based on first-price auction can help here but with this simple auction scheme users report bid amounts lower than their true valuation of the bandwidth resource [11]. VCG mechanism, when either using with optimal allocation algorithms [13], or with approximate allocation algorithm [19], ensures truthfulness but is often computationally intensive to implement in practice. Generalised second price (GSP) auction mechanism, an extension of VCG, is not as computationally intensive as VCG and even though it doesn't guarantee truthfulness, it shares many desirable properties of VCG [11].

We think that in the context of bandwidth reservation for applications in community network clouds, GSP mechanism is a good candidate for using in allocation algorithms that need to run often, i.e. once every second. On the other hand, VCG can be used effectively where users reserve bandwidths over longer periods, for instance a cloud service may want to schedule bandwidth for performing backups every night over the whole month. In such situations, the provider can easily forgo computational efficiency of other approaches for the economic efficiency of the VCG mechanism, as it guarantees maximal social welfare.

5 Conclusion and Future Work

Cloud computing with its success in providing virtualised resources on demand has transformed the technology landscape, revolutionising how Internet applications are developed and delivered to the users. Perhaps now is the right opportunity to take full advantage of the virtualisation and distributed edge model of the cloud computing to design the killer applications for the community network clouds. The success of such cloud applications requires better and guaranteed access to the Internet from within the community network to meet their throughput and latency constraints, for which the current best-effort provisioning model doesn't match up.

In this paper, we applied various pricing mechanisms from the literature to bandwidth allocation problem of cloud applications in community networks, and studied their impact on the social welfare and truthfulness. We found that second-price based GSP and VCG mechanisms are good candidates for arbitration in bandwidth reservation algorithms. However, these mechanism assume that the provider is trustworthy, which in the case of multiple independent providers in the community network with no centralised control is difficult to ensure, and this is the focus for our future work. Another challenge is that multiple users may be connected to the provider using the same path in the community network, and reserving bandwidth for such users in the same time interval may cause congestion across some of the links, which an intelligent allocation algorithm should try to avoid. Lastly, any bandwidth reservation scheme should not negatively impact the normal operation of the community network for its users, so allocation mechanism needs to be adaptive to the network congestion and bandwidth usage in the community network.

Community network clouds build upon the vast research in peer-to-peer, volunteer and edge computing, and we see a huge opportunity in extending this work for building the core community cloud services that drive innovation in many related areas, and not just the edge cloud computing. We believe that the determinant of this success will not be just the technical sophistication with which the research challenges and open problems are solved, but also by how well the enthusiasts of the community network clouds succeed in capturing the imagination and meeting the expectations of the members of the community networks.

Acknowledgement

This work is supported by 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 under contract TIN2013-47245-C2-1-R.

References

1. Baig, R., Roca, R., Freitag, F., Navarro, L.: guifi.net, a crowdsourced network infrastructure held in common. Computer Networks (Jul 2015)
2. Braem, B., et al.: A case for research with and on community networks. ACM SIGCOMM Computer Communication Review 43(3), 68–73 (Jul 2013)
3. Caton, S., Haas, C., Chard, K., Bubendorfer, K., Rana, O.F.: A Social Compute Cloud: Allocating and Sharing Infrastructure Resources via Social Networks. IEEE Transactions on Services Computing 7(3), 359–372 (Jul 2014)

4. Depoorter, W., Vanmechelen, K., Broeckhove, J.: Economic Co-allocation and Advance Reservation of Network and Computational Resources in Grids. In: GECON, pp. 46–60. LNCS, Springer (2012)
5. Gui, Y., Zheng, Z., Wu, F., Gao, X., Chen, G.: SOAR: Strategy-proof auction mechanisms for distributed cloud bandwidth reservation. In: IEEE International Conference on Communication Systems (ICCS 2014). pp. 162–166. IEEE, Macau (Nov 2014)
6. Guo, J., Liu, F., Zeng, D., Lui, J.C.S., Jin, H.: A cooperative game based allocation for sharing data center networks. In: INFOCOM. pp. 2139–2147. IEEE, Turin, Italy (Apr 2013)
7. Jana, R., Kannan, K.N., Yih-Farn Chen, Jana, R., Kannan, K.N.: Using Generalized Second Price Auction for Congestion Pricing. In: GLOBECOM. pp. 1–6. IEEE (Dec 2011)
8. Khan, A.M., Buyuksahin, U.C., Freitag, F.: Towards Incentive-based Resource Assignment and Regulation in Clouds for Community Networks. In: GECON, LNCS, vol. 8193, pp. 197–211. Springer International Publishing (Sep 2013)
9. Khan, A.M., Buyuksahin, U.C., Freitag, F.: Incentive-based Resource Assignment and Regulation for Collaborative Cloud Services in Community Networks. Journal of Computer and System Sciences (Dec 2014)
10. Khan, A.M., Freitag, F.: Sparks in the Fog: Social and Economic Mechanisms as Enablers for Community Network Clouds. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal 3(8) (2014)
11. Maillé, P., Tuffin, B.: Telecommunication Network Economics: From Theory to Applications. Cambridge University Press (2014)
12. Marinos, A., Briscoe, G.: Community Cloud Computing. In: CloudCom, LNCS, vol. 5931, pp. 472–484. Springer Berlin Heidelberg, Beijing, China (Dec 2009)
13. Nisan, N., Ronen, A.: Algorithmic Mechanism Design. In: 31st Annual ACM Symposium on Theory of Computing (STOC 1999). pp. 129–140. ACM Press, New York, New York, USA (Apr 1999)
14. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The Case for VM-Based Cloudlets in Mobile Computing. IEEE Pervasive Computing 8(4), 14–23 (Oct 2009)
15. Shen, H., Li, Z.: New bandwidth sharing and pricing policies to achieve a win-win situation for cloud provider and tenants. In: INFOCOM. pp. 835–843. IEEE, Toronto, Canada (Apr 2014)
16. Vega, D., Meseguer, R., Freitag, F.: Analysis of the Social Effort in Multiplex Participatory Networks. In: GECON, pp. 67–79. LNCS, Springer International Publishing (2014)
17. Wilensky, U.: NetLogo (1999), `http://ccl.northwestern.edu/netlogo/`
18. Xiao, Y., Huang, J., Yuen, C., DaSilva, L.A.: Fairness and efficiency tradeoffs for user cooperation in distributed wireless networks. In: INFOCOM. pp. 285–289. IEEE, Turin, Italy (Apr 2013)
19. Zhang, X., Wu, C., Li, Z., Lau, F.C.M.: A Truthful $(1-\epsilon)$-Optimal Mechanism for On-demand Cloud Resource Provisioning. In: INFOCOM. IEEE (2015)
20. Zheng, Z., Gui, Y., Wu, F., Chen, G.: STAR: Strategy-Proof Double Auctions for Multi-Cloud, Multi-Tenant Bandwidth Reservation. IEEE Transactions on Computers 64(7), 2071 – 2083 (2014)
21. Zhou, H., Leung, K.C., Li, V.O.K.: Auction-based bandwidth allocation and scheduling in noncooperative wireless networks. In: IEEE International Conference on Communications (ICC 2014). pp. 2556–2561. IEEE, Sydney, Australia (Jun 2014)