A Game Theoretical Bandwidth Allocation Mechanism for Cloud Robotics

Lujia WANG, Student Member, IEEE, Max Q.-H. Meng, Fellow, IEEE
Department of Electronic Engineering
The Chinese University of Hong Kong
Shatin N.T., Hong Kong
{ljiang, max}@ee.cuhk.edu.hk

Abstract—Cloud robotics is currently driving interest in both academia and industry, since it would allow robots to off-load computation intensive tasks, combine with multiple robots and even download new skills. Bandwidth allocation is the fundamental and dominant task for resource sharing among users in cloud robotics. However, many technical challenges are still outstanding, since incast congestion happens in high-bandwidth and low-latency networks, when multiple synchronized users send data to a same receiver in parallel [1]. In this paper, we introduce a resource allocation framework for cloud robotics, and propose a game-theoretic problem formulation and linear pricing scheme of bandwidth allocation, we also implement a congestion control algorithm by using optimal parameters derived from the game-theoretic algorithm. Simulation results demonstrate that the proposed mechanism achieves better performance of bandwidth allocation in cloud robotics scenarios.

Index Terms—Cloud Robotics, Resource Allocation, Complete Information Pricing Scheme, Congestion Control

I. INTRODUCTION

Nowadays, there is a growing need of service robots in society, and developing a practical robot that can cover many services would be extremely expensive and take a long time. Instead, it is reasonable to combine multiple robots that have limited abilities, and access vast amounts of processing power and data so as to make a wide variety of abilities and services available. The most effective approach for solving problems arising in service robots so far is the so-called "Cloud Robotics", which combines robot technology with ubiquitous network and cloud-computing infrastructure linking a lot of robots, sensors, portable devices and data center. In [2], researchers at Singapore’s ASORO laboratory have built a cloud computing infrastructure to generate 3-D models of environments, allowing robots to perform simultaneous localization and mapping, which much faster than by relying on their onboard computers, but they do not consider the network latencies or delays and the communication channel’s reliability.

Connecting to the cloud, a robot may solve complex problems that do not require strict real-time performance, however, when the tasks need real-time react, perceive and control mostly out of local computation, the cloud-based solutions can get slow, or simply become unavailable. In order to put cloud robotic system into practice, it is great importance of bandwidth allocation to achieve fair and effective resource allocation in cloud robotics.

Some recent work of resource allocation in cloud computing includes [3]-[5]. Teng and Magoules in [3] introduced the Bayesian Nash Equilibrium Allocation Algorithm in solving resource management problem in cloud computing. Popcorn in [4] utilized virtual currency for users submitting tasks to a centralized server, while Walras [5] used bidding scheme to match the supply and demand. Moreover, some recent work of usage-based pricing and revenue management in communication network includes [6]-[8]. Basar and Srikant in [6] investigated the bandwidth allocation problem in a single link network with the single pricing scheme. Shen and Basar in [7] extended the study to a more general nonlinear pricing case with the incomplete network information scenario. They discussed the single pricing scheme under incomplete information with a continuum distribution of users’ types. Li and Huang in [8] considered the partial differential pricing scheme for resource allocation between service provider and users. However, most studies focus on either services allocation in application layer for service provider, or theoretic resource allocation schemes. Besides, none of the related work proposed a full story of resource allocation implementation on cloud robotics.

In this paper, we introduce a framework of resource allocation in cloud robotics, and study pricing scheme with complete information. According to [9], we choose linear pricing for lower implementing complexity. This scheme is inspired by research [8], we utilize their two stage algorithm and linear pricing with a discrete setting of user’s type which exactly matches our problem. In addition, we modify the algorithm to solve cloud robotics bandwidth allocation in two aspects: at first, we add the congestion cost term into the problem formulation; second, in the admission control stage, we focus on the optimization of admitted accessing user number to improve the queue management and avoid congestion in cloud. Since the nature of traffic and the requirements of applications in cloud are quite different from that in the wide area Internet, especially when multiple synchronized servers send data to a same receiver in parallel, the incast
congestion studied in [10], [11] is common. We also make some modification on current Transmission Control Protocol (TCP) variant DCTCP [12] to improve their performance on congestion control for cloud robotics scenarios.

The rest of the paper is organized as follows. Section 2 introduces the framework of resource allocation for cloud robotics. Section 3 proposes the game-theoretic perspective in formulating and solving the resource allocation problems in cloud robotics. Section 4 presents the CRTCP in implementation congestion control schemes. Section 5 shows the simulation results on performance evaluation. Finally, Section 6 concludes the paper.

II. A FRAMEWORK OF RESOURCE ALLOCATION FOR CLOUD ROBOTICS

In cloud robotics, it is rare that computational resources are allocated totally to a single user. What we expect is that resource will be shared in the cloud datacenter. This section introduces the framework of resource allocation for cloud robotics as shown in Fig.1, it enables heterogeneous robots to sharing knowledge and services that consist of interdependent, power cost tasks.

![Figure 1: Resource Allocation Framework for Cloud Robotics](image1)

It consists of Robot Request Connection Unit (RRCU), Resource Manage Unit (RMU), and Robot Database (RD) that storing information shared among cloud robots and Service Database (SD) provided by some cloud service provider.

The Robot Request Connection Unit classifies the request and provides them different pricing scheme by a pricing menu, then the Resource Manage Unit which access to the cloud does admission control and buffer queue management according to pricing scheme, in the highest level Database Unit stores sharing information and service.

III. FORMULATION THE BANDWIDTH ALLOCATION PROBLEM OF CLOUD ROBOTICS

We model the interaction between Resource Management Unit and robot users as a Stackelberg game [13]. The Resource Management Unit sets the price menu for different resource per bandwidth and robots respond to the price by presenting a certain amount of request flow to the cloud. Fig.2 is an illustrative example of topology between Resource Management Unit and robot users in cloud robotics.

Suppose that a monopolistic cloud Resource Management Unit charges robots for usage of a network to maximize profit. let \( N := \{1, \cdots, n\} \) denote the set of robot users, for robot user \( i \), \( u_i \) is the utility function, \( x_i \) is usage of bandwidth resource and \( p_i \) be the price per unit bandwidth charged by the Resource Manage Unit. Consider link of capacity \( C \) accessed by \( n \) users. For User \( i \) of willingness payment type \( \omega_i \), its net utility [6] is

\[
F_{\omega_i}(x_i, x_{-i}; p_i) = \omega_i \log(1+x_i) - \frac{1}{C - x_i - x_{-i}} - p_i x_i \quad (1)
\]

where User \( i \)'s usage of bandwidth \( x_i \) denote the transmission rate, \( x_{-i} \) is the set of all the other users' usage of bandwidth, \( x_{-i} := \sum_{j=1}^n x_j - x_i \). The logarithmic utility function \( \omega_i \log(1+x_i) \) is verified to ensure non-trivial and meaningful solution to the Stackelberg game as Remark II.1 in [6], and second term is the congestion cost, which captures the delay in the framework of an M/M/1 queue.

![Figure 2: Topology for Cloud Robotics](image2)

While each user tries to maximize its net utility by choosing optimal transmission rate (taking the pricing policy as given), the Resource Manage Unit needs to design optimal pricing policies such that its profit, \( \sum_{i \in N} p_i x_i \), is maximized. Note that what differentiates one user from another
is captured in the relative values of $\omega_i$’s, and “complete information” refers to the situation where the service provider knows the true values of all the $\omega_i$’s. Therefore, with complete information the Resource Manage Unit can distinguish different groups of users, it can classify what kind of request it is, and announces the pricing and the admission control decisions to different robot users.

A. User’s Surplus Maximization Problem

User $i$ determines $x_i$ to maximize its utility and the problem is given by

$$\max_{0 \leq x_i \leq n - x_{i}^*} F_i(x_i, x_{-i}^*; p_i)$$

subject to

$$\sum_{i \in K} x_i \leq C$$

From the first order necessary conditions (which are also sufficient) for a Nash equilibrium among the followers, User $i$’s price can be denoted as

$$p_i = \frac{\omega_i}{1 + x_i} - \frac{1}{(C - \bar{x})^2}$$

where $\bar{x} = \sum_{j=1}^n x_j$.

B. RMU’s Pricing and Admission Control Problem

The RMU maximizes its revenue by choosing the price $p_i$ and the admitted user number $K$ for the limited bandwidth. Then the problem is given by

$$\max_{p_i \geq 0, y_i \in \{0, 1\}^K} \sum_{i \in K} y_i L(p_i, x_i)$$

subject to

$$p_i = \frac{\omega_i}{1 + x_i}, \quad i \in N$$

$$K \in \{0, \cdots, N\}$$

$$\sum_{i \in K} x_i \leq C$$

where $L(p_i, x_i) = p_i x_i$, $y_i \in \{0, 1\}^K$ means $y_i$ equals to 0 after $i$ is larger than $K$ and we put the congestion cost $\frac{1}{(C - \bar{x})^2}$ in $p_i$ as a constraint $\sum_{i \in K} x_i \leq C$, or else the congestion control would be considered twice in this problem.

What we have here is therefore a Stackelberg game, with one leader (with objective function $L$) and $n$ noncooperative Nash followers (with objective functions $F_i$). Then the solution algorithm can be formulated as the following two steps:

Step one is resource allocation. For a fixed admitted user number $K$, solve the optimal transmission rate $x$. Plugging (6) into (4), the above non-convex optimization problem be easily converted to convex as follows.

$$\max_{p_i \geq 0, x_i \geq 0, K} \left( \frac{\omega_i x_i}{1 + x_i} \right)$$

subject to

$$\sum_{i \in K} x_i \leq C$$

Since the revenue is strictly concave, and the constraint set is convex and compact, this optimization problem admits a optimal solution for transmission rates of users, and hence lead to a unique allocation. By using the Lagrange multiplier technique, we can get the optimal transmission rate that denoted as

$$x_i^* = \sqrt{\frac{\omega_i}{\lambda}} - 1$$

where $\lambda$ be the Lagrange multiplier corresponding to the bandwidth constraints (9).

Step two is admission control. We note the bandwidth constraints (9) must hold equality, since the objective is strictly increasing function in $x_i$. Thus, by plugging (10) into (9), we have

$$\sum_{i \in K} (\sqrt{\frac{\omega_i}{\lambda}} - 1) = C$$

Assuming that, $\omega_1 \geq \omega_2 \geq \cdots \geq \omega_N$, then $\lambda^*$ must satisfy the above condition (11). For a admitted user number threshold value $K_{th}$ satisfying

$$\frac{\omega_{K_{th}}}{\lambda^*} \geq 1$$

$$\frac{\omega_{K_{th} + 1}}{\lambda^*} \leq 1$$

where $K_{th}$ is going to used as admission control threshold and only user with index no larger than $K_{th}$ will be allocated the positive bandwidth.

Algorithm1 Sloving the Revenue Maximization Problem

1: function Revenue($i, \omega_i, C, i \in N$)
2: $k \leftarrow N, \lambda(k) \leftarrow (\frac{\omega_k}{\sqrt{\omega_{K_{th} + 1}}} + C_{th})^2$
3: while $\omega_k \leq \lambda(k)$ do
4: $k \leftarrow k + 1, \lambda(k) \leftarrow (\frac{\omega_k}{\sqrt{\omega_{K_{th} + 1}}} + C_{th})^2$
5: end while
6: $K_{th} \leftarrow k, \lambda^* \leftarrow \lambda(k)$
7: return $K_{th}, \lambda^*$
8: end function

With $K_{th}$ and $\lambda^*$, the above Surplus and Revenue Maximization problems can be solved according to the following Theorem which lead to an optimal solution.

Theorem. There exist optimal solution of Resource Allocation Problem that satisfies the following conditions:

• Under the limited bandwidth resources $K_{th}$ users are admitted into the queue: $n^* = K_{th}$.

• There exists a value $\lambda^*$ with the above admitted user number $K_{th} \leq N$, such that each User $i$ receive an optimal bandwidth:

$$x_i^* = \begin{cases} \sqrt{\frac{\omega_i}{\lambda}} - 1, & i = 1, \cdots, K_{th}; \\ 0, & \text{otherwise.} \end{cases}$$

with the price

$$p_i^* = \begin{cases} \sqrt{\omega_i \lambda}, & i = 1, \cdots, K_{th}; \\ \omega_i, & \text{otherwise.} \end{cases}$$
The value of $\lambda^*$ and $K_{th}$ can be computed as in Algorithm 1, for all $i \in N$.

IV. IMPLEMENTATION

A. Congestion Schemes for Cloud Robotics

Since cloud robotics may generate a mixture of bursty query traffic, delay-sensitive short messages, and throughput-intensive long flows that often have strict deadlines for the completion. We make some modifications on the current Transmission Control Protocol (TCP) variant DCTCP [12] which is developed to solve incast congestion for datacenter, and improve its performance on congestion control for cloud robotics scenario. We make use of their main algorithm: a simple active queue management based on Explicit Congestion Notification (ECN) [14] at the switch, and window control scheme at the source which reacts to ECN marks by reducing window size in proportion to fraction of packets that are marked.

Furthermore, we improve the algorithm to solve cloud robotics bandwidth allocation in the following aspects: in the active queue management, we use the optimal admitted threshold $K_{th}$ derived from Algorithm 1 for queue management; in the window control scheme, we change the control interval from RTT to $2^*\text{RTT}$ to make the window control more smoothly and steady. We call it cloud robotics Transmission Control Protocol (CRTCP) for better represent in the simulation comparison, and its implementation details is as below:

At first, a very simple active queue management scheme is adopted at the switch side. When a packet arrives at the switch buffer and the buffer occupancy is at least $K_{th}$ packets, where it is marked using the Explicit Congestion Notification (ECN) mechanism.

Second, to achieve high throughput and very low queue occupancies, the source reduces its window by a factor that is proportional to the fraction of marked packets; the larger the fraction, the larger the decrease factor. The source rate adaptation model is described in differential equation form as:

$$\frac{dW}{dt} = \frac{x(t)}{W(t)} - \frac{W(t)\alpha(t)}{2} x(t)p(t - R^*)$$  \hspace{1cm} (16)

$$\frac{d\alpha}{dt} = \frac{x(t)u}{W(t)} (p(t - R^*) - \alpha(t))$$  \hspace{1cm} (17)

where $p(t)$ indicates the packet marking process at the switch and is given by

$$p(t) = 1_{q(t) \geq K_{th}}$$  \hspace{1cm} (18)

Equation (16) models the window control scheme and consists of standard additive increase term $\frac{x(t)}{W(t)}$, and a multiplicative decrease term $\frac{W(t)\alpha(t)}{2} x(t)p(t - R^*)$ which models the source’s reduction of its window size by a factor of $\alpha(t)/2$ when packets are marked, and this occurs once two

Round-trip Time (RTT). Moreover, we use a fixed value $R^* = d + \frac{K_{th}}{2}$ for the delay, where $d$ is propagation delay by assuming that it is equal for all flows, and $K_{th}/C$ is queueing delay.

B. Bandwidth Allocation Implementation Algorithm in Cloud Robotics

Algorithm 2 explains how the Stackelberg game combined with congestion control scheme allocating bandwidth in cloud robotics. Resource Manage Unit first initialize available bandwidth, then it generates a provision information and published it to users. At the same time, cloud users (or robots) request services in terms of their own demand and willingness payment. According to the admitted user number, Resource Manage Unit marks the ACK when more than $K_{th}$ users request and update fraction $F$ for every ACK as follow [12]:

$$\alpha \leftarrow (1 - u)\alpha + uF$$  \hspace{1cm} (19)

where $F$ is the fraction of ACK that were marked in the most recent window of data, and $u \in (0, 1)$ is a fixed parameter. Moreover, we use $\alpha$ to cut the window size to response to a marked ACK as follows [12]:

$$W \leftarrow (1 - \frac{\alpha}{2})W$$  \hspace{1cm} (20)

Note that to estimate the throughput of a TCP connection for receive window adjustment, the shortest time scale is an RTT for that connection. However, the control interval for a TCP connection is $2^*\text{RTT}$ in our algorithm, since we need one RTT latency for that adjusted window to take effect which is inspired by [10]. Every two RTT, the scheme in RMU checks users’ demand and payment, and adjusts pricing menu and window size.

Algorithm 2 Bandwidth Allocation Implementation with TCP

1: Initialize
2: RMU check: user number $N \geq 2$, available bandwidth and buffer length.
3: cloud users submit request and willingness payment
4: datacenter virtualizes the resources according to request
5: RMU announces the price $p^*$

6: for every two RTT, do
7: \hspace{1cm} $(K_{th}, \lambda^*, \alpha) = \text{Revenue}(i, \omega_i, C, \lambda(i))$
8: \hspace{1cm} calculates $x^*$, $p^*$
9: \hspace{1cm} if $q(t) \geq K_{th}$
10: \hspace{1cm} marks the ACK; ECN = 1
11: \hspace{1cm} $F = \frac{\text{marked ACKs}}{\text{total ACKs}}$
12: \hspace{1cm} $\alpha \leftarrow (1 - u)\alpha + uF$
13: \hspace{1cm} $W \leftarrow (1 - \frac{\alpha}{2})W$
14: \hspace{1cm} return $\alpha, W$
15: \hspace{1cm} else
16: \hspace{1cm} $\text{ECN} = 0$
17: \hspace{1cm} $W += \frac{1}{W}$
18: \hspace{1cm} end if
19: end for
V. PERFORMANCE EVALUATION

Using the above congestion control scheme developed, we have studied the average optimum throughputs, average price levels, revenue per unit bandwidth, and user's net utility levels for resource allocation in cloud robotics. Moreover, we have also corroborated our results with ns2 simulations.

A. Cloud Topology

To evaluate performance of the proposed congestion control scheme for cloud robotics, the basic network structure as shown in Fig.3, it captures cloud features, such as many-to-many, many-to-one and multi-bottleneck environment. There are two kinds of traffic pattern in the topology: traffic from sources \( \{s_1, \ldots, s_n\} \) to destination A \( \{d_0\} \) are many-to-one pattern, and traffic from sources \( \{s_1, \ldots, s_n\} \) to destination B \( \{d_1, \ldots, d_n\} \) are many-to-many pattern. Especially, we set multi-bottleneck as \( r_1 \rightarrow r_2 \rightarrow r_3 \) in the many-to-many pattern. The parameters used are: bottleneck delay is 200 bps, delay of side links \( d = 100\mu s \), bandwidth of side links is 1Gbps, while bandwidth between bottleneck links is 10Gbps. Furthermore, pricing and utility parameters are fixed for a given simulation.

![Fig. 3. Topology of Cloud](image)

B. Optimum Values Analysis

Consider the fixed willingness payment difference is \( \omega_{i+1} - \omega_i = 0.05 \), compute the Lagrange multiplier \( \lambda^* \), average optimum throughputs \( \bar{x}^* \), average price levels \( \bar{p}^* \), and total user number \( N \) using MATLAB. Table I indicates that \( x^* \) increasing when \( p^* \) decreasing, the permitted number threshold increasing when the Lagrange multiplier decreasing, thus it demonstrates an overall monotonic trend. Moreover, we use ns2 simulations to evaluate the throughput achieved as the marking threshold \( K_{th} \) is varied from 1% to 25% of the BDP. Parameters are: bandwidth-delay product is 400 packets, and \( u = 0.05 \). We choose the many-to-one traffic pattern and consider two cases with \( N = 5 \) and \( N = 25 \) long-lived flows. The simulations results, as shown in Fig.4, clearly show that CRTCP achieves high throughput, even with marking threshold as low as 1% of the bandwidth-delay product. However, throughput of TCP improves as the number of flows increase.

C. Incast Scenario Analysis

We simulate two incast scenario as mentioned before and the throughput versus sender numbers, as shown in Fig.5 and Fig.6, are the average results of 10 simulation rounds.

At first, in the many-to-one scenario, \( n \) source nodes generate the same amount of traffic to a specific destination \( D_0 \) through one bottleneck link \( r_1 \) to \( r_2 \). We set \( K_{th} = 65 \) packets, and \( u = 1/16 (K_{th} \) and \( u \) are chosen as [12] to match the 10Gbps bottleneck bandwidth). As shown in Fig.5, the CRTCP achieves smooth and increasing throughput with the number of source nodes increasing. Larger data amount per source node results in slightly higher throughput achieved. However, the throughput of TCP before incast congestion is actually higher than CRTCP, it begins to suffer timeout when the number of senders exceeds 10.

Second, in the many-to-many scenario, \( n \) source nodes generate traffic to Destination B of \( n \) receiver nodes through two bottleneck links of \( r_1 \) to \( r_2 \) and \( r_2 \) to \( r_3 \). There are multiple simultaneous incasts going on, as occurs in the production clusters. In other words, \( n \) incast simulations happen all at once. With the total data volume of all senders is fixed and the number of senders varies, throughput versus the number of senders as shown in Fig.5, we observe that CRTCP keeps the demand on buffer space low enough that the dynamic buffering is able to cover all requests for memory.

![Fig. 4. Throughput vs Marking Threshold \( K_{th} \)](image)

<table>
<thead>
<tr>
<th>( K_{th} )</th>
<th>( \lambda^* )</th>
<th>( \bar{x}^* )</th>
<th>( \bar{p}^* )</th>
<th>( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.7229</td>
<td>0.5218</td>
<td>0.2750</td>
<td>8</td>
</tr>
<tr>
<td>20</td>
<td>0.3810</td>
<td>0.6337</td>
<td>0.7750</td>
<td>25</td>
</tr>
<tr>
<td>45</td>
<td>0.3248</td>
<td>0.7316</td>
<td>1.2750</td>
<td>50</td>
</tr>
<tr>
<td>65</td>
<td>0.1778</td>
<td>1.0489</td>
<td>1.7750</td>
<td>70</td>
</tr>
</tbody>
</table>

### Table I

RESULTS COMPARISON OF OPTIMUM VALUES
and suffers no timeouts at all. On the other hand, TCP performs poorly, over 55% of the requests suffer from at least one timeout.

VI. CONCLUSION

In this paper, we have presented the design, implementation, and evaluation of resource allocation in cloud robotics, especially to improve performance of bandwidth allocation under the high bandwidth and low-latency operation condition in cloud. In admission control and congestion control during bandwidth allocation, different from approaches to using the "optimization-orientated" pricing for network resource allocation, we focus on the "economics-based" pricing, and propose a linear pricing algorithm which lead to user’s surplus and RMU’s revenue maximization by building a Stackelberg game. Moreover, Transmission Control Protocol is modified by using the derived optimal parameter value in its implementation, and simulation results confirm the theoretical analysis very well. Furthermore, in our pricing scheme we assume that the RMU knows all information of users such as their willingness payment, however, it is not always true in practice. Therefore, the proposed pricing scheme need more investigation under incomplete information in our future work.

ACKNOWLEDGMENT

The authors would like to thank Rui Zhang and Baopu Li who gave constructive suggestions on an early draft of this paper.

REFERENCES


