

# Poster Abstract: API QoS Prediction for Apps in Cellular Networks

Yali Zhao, Shangguang Wang\*, Lubao Wang  
State Key Laboratory of Networking and Switching Technology  
Beijing University of Posts and Telecommunications  
Beijing, China  
{zhaoyali;sgwang;lubao}@bupt.edu.cn

**Abstract**—One App running on mobile devices often needs to invoke several frequently-used Application Programming Interfaces (APIs) to perform service provision. However, when these mobile devices roam around another city, these APIs' QoS performance changes or degradation often make App failure. Hence, accurate QoS prediction before these APIs are invoked becomes an important issue for App developers. In this paper, we propose an accurate API QoS prediction approach by using user similarity computation and base-station similarity computing. The experimental results demonstrate the efficiency and effectiveness of our approach.

**Keywords**—API; App; QoS; base-station

## I. INTRODUCTION

With the rapid development of smart phones and other mobile devices, the Internet is no longer confined to PC. More and more smart mobile devices invoked the Internet by connecting base-stations which have been built everywhere, forming cellular networks. Based on the growing popularity of mobile devices, a large number of Apps have been developed and Apps running on mobile devices often need to invoke multiple Application Programming Interfaces (APIs) to perform service provision by connecting base-stations of cellular networks. Then, it is very important to know which APIs have better QoS (values) for Apps performance optimization. Hence, how to accurately predict the QoS before APIs are invoked is a very important issue for Apps performance optimization of mobile devices.

The API QoS are notably more volatile, and mobile devices are often roaming around in cellular networks. Due to the mobility of mobile devices, history QoS of APIs in the past base-station will fail when mobile devices move around another city and the API QoS in the current base-station is empty. Note that in order to easily understand our approach, we take user represent mobile device in this paper. Although many QoS prediction approaches have been proposed in Internet environments, but they often fail in making accurate API QoS prediction in cellular networks [1]. For instance, for one user called Sam in Beijing, when use one video App on his mobile phone, the App will invoke one (video compression encoding) API in cellular network environments, and then its response time is 100 msec on average where the host server running the API is deployed in Beijing. When the user roam around Shanghai, if the App still use the API, traditional prediction approaches often monitor its historical QoS data of Beijing and obtained response time is still 100 msec. However, its real response time is higher than 100 msec which leads a

sharp drop in App QoS performance, or even App invoking failure. The main reason why traditional approaches fail in accurate QoS prediction is that they do not take user mobility into account [2]. When one Beijing user travels to Shanghai, the cellular network environment has changed and the history QoS data of the API in Beijing is invalid for the Shanghai user.

Different from traditional approaches, when users roam around another city, if there are some users in the same base-station, they invoked the API, then we can predict the QoS based on their historical data; otherwise, we use other users' historical data from other base-stations where they invoked the API.

## II. OUR APPROACH

In this approach, we firstly calculate similar base-stations by adopting Pearson Correlation Coefficient or find similar users, and then select Top-K users or base-stations to predict API QoS.

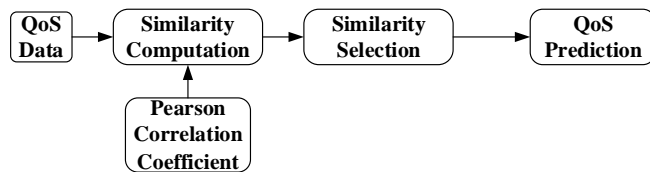


Figure 1. Procedure of our approach

### A. Similarity Computation

When we predict the QoS of API  $a$  invoked by the active user  $u$ , we must take the current base-station of the active user  $u$  into account. According to the API  $a$  invoked by user  $u$ , we can divide the situation into two cases, as follows:

**Case 1.** The API  $a$  invoked by the active user  $u$  has history API QoS in the current base-station, i.e., users in control of the base-station adopted the same API as the active user  $u$  before, so the history QoS of the API  $a$  is stored in the current base-station.

**Case 2.** The API  $a$  invoked by the active user  $u$  has no history QoS in the current base-station, i.e., users in control of the base-station did not adopt the API  $a$  before, so there is no history QoS of the API  $a$ .

#### 1) User Similarity Computation

For situation as Case 1 described, the history API QoS in the past base-station of the active user  $u$  is invalid and it cannot be used for user similarity computation. However, we can use the history QoS of the same API as the active user  $u$  adopts.

More and more base-stations are built and the distribution is concentrated. Based on the condition, we think users in the current base-station who adopted the API  $a$  are similar.

The similar users set of active user  $u$  is described as  $S_u(u_i) = \{u_1, u_2, \dots, u_i, \dots, U\}$ ,  $i = 1, 2, \dots, U$ , where  $U$  denotes the total number users adopted the API  $a$  in the current base-station.

## 2) Base-Station Similarity Computation

In this paper, we assume  $q_{u,a}^t$  represents the history API QoS of user  $u$  repeatedly invokes API  $a$  ( $a=1,2,3, \dots$ ) at the  $t$ -th ( $t=1,2,3, \dots$ ) time.

For situation as Case 2 described, i.e., the API  $a$  invoked by the active user  $u$  has no history API QoS in the current base-station. We should find the similar base-stations for it. Based on the Pearson Correlation Coefficient (PCC), base-station similarity computation employs the similarity between base-stations. We calculate the similarity between base-station  $b_1$  and base-station  $b_2$  by the following:

$$\text{sim}_{b_1, b_2} = \frac{1}{A} \sum_{a=1}^A \frac{\sum_{u=1}^U (P_{b_1, a}^u - E_{b_1, a})(P_{b_2, a}^u - E_{b_2, a})}{\sqrt{\sum_{u=1}^U (P_{b_1, a}^u - E_{b_1, a})^2} \sqrt{\sum_{u=1}^U (P_{b_2, a}^u - E_{b_2, a})^2}} \quad (1)$$

$$\text{with } P_{b_1, a}^u = \frac{\sum_{t=1}^T q_{u, a}^t}{T} \quad (2)$$

$$E_{b_1, a} = \frac{\sum_{u=1}^U P_{b_1, a}^u}{U} \quad (3)$$

where  $\text{sim}_{b_1, b_2}$  is the similarity between base-station  $b_1$  and base-station  $b_2$ .  $P_{b_1, a}^u$  represents the QoS of API  $a$  invoked by user  $u$  in the base-station  $b_1$ .  $E_{b_1, a}$  represents the QoS expectation of API  $a$  invoked in the base-station  $b_1$ . Where  $A$  denotes the number of same APIs that invoked in base-station  $b_1$  and  $b_2$ .

If two base-stations happen to have similar QoS experience on a few same APIs invoked, then using the PCC will overestimate the similarities of base-stations. To address this problem, we employ a significance weight to reduce the influence of a small number of similar APIs invoked [3]. An enhanced PCC between different base-stations is defined as follows:

$$\text{sim}'_{b_1, b_2} = \frac{2 * |A_{b_1} \cap A_{b_2}|}{|A_{b_1}| + |A_{b_2}|} \text{sim}_{b_1, b_2} \quad (4)$$

where  $|A_{b_1} \cap A_{b_2}|$  is the number of APIs that invoked both in the base-station  $b_1$  and the base-station  $b_2$ , and  $|A_{b_1}|$  and  $|A_{b_2}|$  are the number of APIs invoked in the base-station  $b_1$  and base-station  $b_2$  respectively.

## B. Top-K Similarity Selection

We select Top-K similar users based on the distance between other users and the active user. The shorter the distance, the stronger the similarity. The Top-K similar users set of active user  $u$  as  $S'_u(u_i) = \{u_i \in S_u(u_i), \text{dist}(u, u_i) >$

$0, u \neq u_i\}$ , where  $\text{dist}(u, u_i)$  represents the distance between  $u$  and  $u_i$  ( $i = 1, 2, \dots, K$ ).

We use the Top-K to rank the base-stations based on PCC similarities and select the Top-K most similar base-stations for making value prediction. A set of Top-K similar base-stations of base-station  $b$  can be found as  $S_b(b_i) = \{b_i | \text{sim}'_{b, b_i} > 0, b_i \neq b, i = 1, 2, \dots, K\}$ .

## C. QoS Prediction

According to whether the QoS of API invoked by active users exist history data in the current base-station, we predict the API QoS for active users as following:

- 1) Based on the user similarity, we predict the API QoS of the active user  $u$  as follows:

$$\text{pred}_{user}(u, a) = \frac{1}{K} \sum_{u_i \in S'_u(u_i)} \frac{\sum_{t=1}^T q_{u_i, a}^t}{T} \quad (5)$$

where  $q_{u_i, a}^t$  represents the QoS of API  $a$  invoked by user  $u_i$  at the  $t$ -th time,  $T$  is the total times that the API  $a$  invoked by user  $u_i$  ( $i = 1, 2, \dots, K$ ) repeatedly.

- 2) Based on the Top-K similar base-stations, we proposed an approach to predict the API QoS for the API  $a$  invoked by active user  $u$  as follows:

$$\text{pred}_{base-station}(u, a) = \frac{\sum_{b_i \in S_b(b_i)} \text{sim}'_{b, b_i} * E_{b_i, a}}{\sum_{b_i \in S_b(b_i)} \text{sim}'_{b, b_i}} \quad (6)$$

where  $E_{b_i, a}$  represents the QoS expectation of API  $a$  invoked in the base-station  $b_i$  ( $i = 1, 2, \dots, K$ ) and  $b$  represents the current base-station of the active user  $u$ .

## III. EXPERIMENT AND CONCLUSION

We conducted simulation-based experiment with the NS-3 simulator<sup>1</sup>. We simulated mobile devices and base-stations by the LTE module. The experimental results show the efficiency and effectiveness of our approach. In this paper, we presented an approach to predict the new APIs' QoS of base-station with consideration of users' mobility and the volatile of API QoS. Compared to previous approaches, our approach considers not only the volatile of API QoS but also users' mobility to adapt the mobile environment. More than that, we have proposed two different approaches according to whether the related QoS in the current base-station.

## REFERENCES

- [1] Y. Ma, S. Wang, P. C. Hung, C.-H. Hsu, Q. Sun, and F. Yang, "A highly accurate prediction algorithm for unknown web service QoS values," *IEEE Transactions on Services Computing*, 2015, DOI: 10.1109/TSC.2015.2407877.
- [2] Y. Ma, S. Wang, F. Yang, and R. N. Chang, "Predicting QoS Values via Multi-Dimensional QoS Data for Web Service Recommendations," *In Proceedings of IEEE International Conference on Web Services (ICWS 2015)*, 2015, pp. 249-256.
- [3] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Transactions on Services Computing*, no.2, vol. 4, pp. 140-152, 2011.

<sup>1</sup> <https://www.nsnam.org>