

A Replay Approach for Remote Testing User Experience of Mobile Bursty Data Application

<http://dx.doi.org/10.3991/ijoe.v9iS7.3187>

Lei Chen¹, Ping Wang¹, Fuqiang Liu¹, Chao Wang¹, Haibo Zhou², Lijun Pu², Jiping Xiong³ and Nguyen Ngoc Van⁴

¹ Broadband Wireless Communications and Multimedia Laboratory, Key Laboratory of Embedded System and Service Computing supported by Ministry of Education, Tongji University, Shanghai, China

² Huawei Technologies Co. Ltd., Shenzhen, China

³ Zhejiang Normal University, Jinhua, China

⁴ Hanoi University of Science and Technology, Hanoi, Vietnam

Abstract—The Bursty data applications become popular in mobile networks, and the quality of end user experience of these applications is vital for mobile telecom operators. Mobile telecom operators and equipment providers begin to design new scheduling strategies aiming at improving quality of user experience. These scheduling strategies can identify particular applications and assign corresponding scheme to them. But traditional test approaches only consider the common performance indicators, so they cannot evaluate the new scheduling strategies. This paper proposes an approach of replaying user behaviors to evaluate the communication system. The approach uses user behavior model to build test scenarios, and evaluate the performance of each application via corresponding particular indicators. Finally, a remote test system is built, and the experiment on this system proves the effectiveness of this approach.

Index Terms—A Quality of user experience, bursty data application, mobile communication

I. INTRODUCTION

The bursty data applications include Social Network Service (SNS), Instant Message (IM), Email, and so on. The bursty data applications in this paper only refer to applications with a small amount of data flow. Applications with continuous large data flow, such as video and audio applications, are excluded. These bursty data applications become more and more popular in mobile networks. This trend motivates telecom operators and equipment providers to design new scheduling strategies for guaranteeing the quality of user experience (QoE) and saving resources [1-3]. These smart scheduling strategies can recognize particular applications depending on deep packet inspection (DPI), and then assign particular schemes to corresponding applications to improve QoE. This brings challenges to traditional test systems and emulators. Traditional test systems evaluate the performance of communication systems by using common indicators, and build scenarios by traffic model of the data flow of all applications [4-7]. But new smart scheduling strategies distinguish different applications so that each application should have its own QoE indicators and traffic model.

A. Related Work

A test system should be able to activate data flow to simulate different application scenarios of real world, and evaluate the performance of communication systems. Traditional test systems activate data flow and build scenarios according to traffic model. Traffic model usually characterizes the fluctuation of overall data flow [8-9]. To evaluate new scheduling strategies, however, test systems must consider the character of each independent data flow of every application. And most traditional test systems send random data so that they cannot test the ability of scheduling program to recognize the particular application. So new test system need to send real data. Reference [10] proposed a mechanism to play back real data for improving simulation accuracy. But this method must capture the overall data flow from every scenario that people want to simulate.

Another important problem is to design performance indicators from the viewpoint of user experience. ITU-T has proposed the criterion for web QoE evaluation [15]. But most bursty data applications cannot be referred simply as web browsing. And many researches focus on how to identify Key Quality Indicator (KQIs) for different bursty data applications. According to a rough classification in [16], bursty data applications belong to transactions-oriented applications characterized by request/response data flow corresponding to bidirectional data transfers, so the user experience is mainly related to the delay of the answer to a request. [17-18] give steps to identify the KQIs. [19] gives a case for developing the common KQIs of data application on commercial mobile network. But lower layer KPIs cannot accurately reflect the QoE, and common KQIs do not distinguish various applications.

B. Contributions

This paper proposes a new approach to evaluate various bursty data applications via their own particular performance indicators. This approach replays real data collected from commercial networks, and we also propose a criterion to identify particular KQIs of various applications and give a template to compute the KQIs. Then we survey the character of user behavior of busy data applications, and find that most data flows only belong to a few typical user actions. Finally we implement

a test system to remotely evaluate the performance of bursty data applications on mobile network.

The remaining of this paper is organized as follows. Section 2 describes the proposed approach. Experiment results and analysis are presented in Section 3. Finally, a conclusion is made in Section 4.

II. REPLAY APPROACH

This section describes the working process of the proposed replay approach and the architecture of the test system. As shown in Fig. 1, the replay approach consists of three stages:

(1) KQIs identification: The tasks of this stage are identifying the KQIs for each application from the viewpoint of QoE and designing the method to compute the values of KQIs.

(2) Modeling user behavior: The aim of modeling user behavior is to build scenario according user behavior model, and then design test cases.

(3) Replay and KQIs analysis: The last step is to run test case in the test system and analyze the results. The architecture of the test system based on replaying real data is given in Section 2.3, where a test system is implemented.

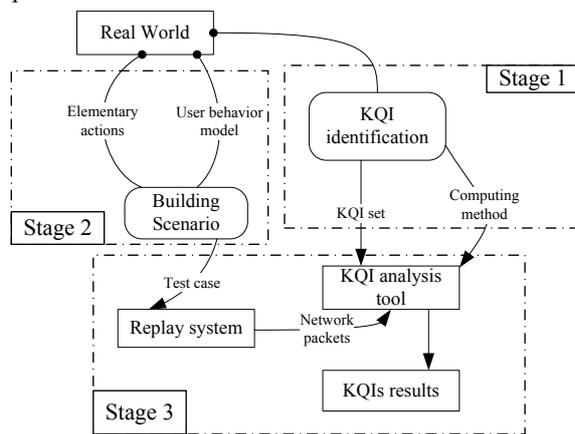


Figure 1. Three stages of the replay approach

A. KQIs Identification

1. KQIs identification for various applications

From QoE's viewpoint, each application has its own user actions and software actions. Some of the actions influence QoE. Some do not. Firstly, we need to design particular KQIs for various applications. The delay and success probabilities of trip-round in the application are important for user experience. We divide the KQIs of bursty data applications into three classes as follows: (1) The delay and success probabilities of some actions can be perceived by users, e.g., the delay of opening a web site, and that of submitting a comment on blog. (2) The delay of some actions cannot be directly perceived by users, but a long delay can cause confusion to users. For instance, when two persons chat using IM applications, they normally do not perceive the delay of sending messages. However, long delay may cause the messages to arrive at the receivers in wrong orders and thus lead to confusion to the users. (3) Most automatic software actions cannot be perceived directly by users. But some of them can also

influence user experience. For example, many applications send a heartbeat packets to keep online, the failure of heartbeat arriving to server will cause offline and then cause negative user experience.

Another principle of designing KQIs is that the KQIs should only concern the whole action which can be perceived by users. This means that sub-processes should be excluded. For example, although the action of authentication in a login process can influence QoE, the delay of authentication should not be a KQI, because a user perceives only the whole delay of login.

2. KQIs Computation

To compute the KQIs of each application, the test system must recognize every action concerning the KQIs, and then match the first and last packets belonging to the same execution of an action. To complete this computation, we must synthesize the information from different layers of network. The actions of applications are quite different so that it is difficult to design a common method to recognize and compute KQIs of various applications. Therefore, we propose a template consisting of three forms to compute KQIs. The form 1, namely attribute form, defines the key attributes of each data packet which are useful to compute KQIs. It has two columns. The first column lists the serial number of the attribute. If the attribute is a common attribute of network policy, give its name in the second column. If the attribute is a particular key data field of this application, give the location or starting-ending characters of this data field. The form 2, namely packet form, gives some context-free conditions and context conditions to recognize the key packets according to the attributes. The context-free conditions describe the key packets' own characters. The context conditions describe the relationship between different packets, since some key packets cannot be recognized by using only their own characters. Finally, KQI form, i.e. form 3, gives formulas to compute KQI on the basis of key packets and the constraints are given to match the related packets.

Table 1 illustrates a simple example of using this template to compute the delay and the success probabilities of sending messages using QQ, which is the most popular IM application in China.

TABLE I.
COMPUTING QQ MESSAGE DELAY AND SUCCESS PROBABILITIES

Form 1.				
Attribute No.	Value			
att. 1	source IP			
att. 2	4th and 5th byte of application layer			
att. 3	6th and 7th byte of application layer			
att. 4	time stamp			
att. 5	Destination IP			
Form 2.				
Action	Criteria	Context Free Condition		Context Condition
		NO.	Condition	
Send	A & B	A	att. 1 in a particular interval	
		B	att 2= '0x00cd'	
Respo-nse	A & B	A	att. 5 in a particular interval	
		B	att. 2= '0x00cd'	

Form 3.		
KOI	formula	constraint
Send success	(amount of response) / (amount of send)	
Send delay	(response.att 4) - (send.att 4)	response.att3 =send.att3
		value < threshold

The form 1 has 5 attributes including IP, time, and two key data fields in QQ application layer data. Form 2 shows that if the attribute 2 is "0x00cd," the packet is a sending or a responding message of QQ. According to the IP, we can distinguish sending and responding messages. Here there is no context condition. Form 3 defines two KOIs. The 'success' indicator means success probability which is equal to the ratio of the amount of responding message over the amount of sending message. The 'send delay' is the delay between responding and sending messages. In the constraint column, attribute 3 of the packets determines which responding messages match which sending messages, and the delay should be less than a threshold.

B. Modeling User Behavior

Figure axis labels are often a source of confusion. Use words rather than symbols. For example, write "Magnetization," or "Magnetization, M," not just "M." Put units in parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization ($A \cdot m^{-1}$)." Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)," not "Temperature/K."

From the viewpoint of QoE, we focus on the performance of user and software actions which can influence QoE, not the simple round-trip process in network layer. So the test system must activate real user and software actions in communication system, and build a test scenario using user behavior model instead of traffic model.

User behavior model concerns three factors: action sequence, the size of the user data field, and the frequency of the action. Much research addressing user behavior shows the following facts. (1) For many applications, there are several typical action sequences. (2) Most data flows of an application concern only a few kinds of user and software actions. (3) The size of user data field and the frequency of the main user action have a concentrated distribution. For example, [20] shows that 12 kinds of user actions cause more than 99% HTTP request/response data flow in FACEBOOK, the biggest social network application in the world.

We extract the network data packet of every user from a base station and then recognize all QQ data flow. The results show that "update friends information", "look up comment", "heart beat" and "send/receive message" generate more than 60% packets in all the QQ data packets of 261 QQ users in 2 hours. Fig. 2 shows the statistic result of 60,000 blogs we randomly selected in SINA Micro Blog, which is the most popular blog application in China. The length of the blog has an obvious concentrated distribution.

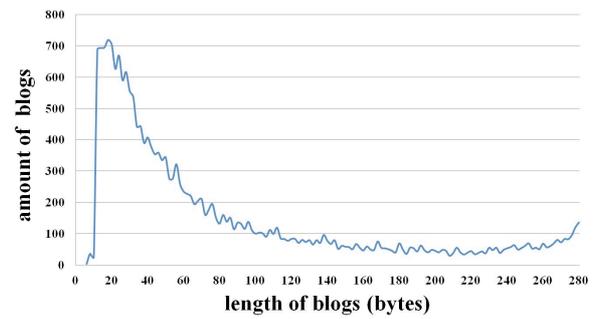


Figure 2. Distribution of the length of SINA Micro Blog

We conclude that complex scenarios can be built by using a few kinds of typical real data packets according to a good user behavior model. And researchers have proposed several effective approaches to model user behavior. "Click stream analysis" [11-12] can be used to find typical action sequence and action frequency of each application. "Web crawler" [13-14] can be used to analyze web applications. Finally, because most IM applications encrypt the user data field, collecting log from volunteers is also an important method to analyze the size of user data field.

As a result, we can build a typical action warehouse. Each action in it consists of real data packets captured from network when this action happens. And then these actions can compose user behavior in various scenarios. One user behavior is a test case. We design a simple script language to edit test case, which covers three essentials: user data length, action frequency and action sequence. An example is shown as follows.

```
QQ_frequent.T:
wait 10000
import \QQ\send_10_characters.pcap
wait 10000
import \QQ\receive_10_characters.pcap
```

```
QQ.T:
import \QQ\QQlogin.pcap
loop 50
import QQ_frequent.T
end loop
import \QQ\QQlogout.pcap
```

The "pcap" file includes the network data packets activated by a action, the T file is a script to describe the application scenario. In "QQ_frequent.T", a user sends and receives a 10-character message every 10 seconds. In "Q.T", a user logs in QQ, does actions of "QQ_frequent.T" 50 times, and then finally logs out. The 'import' sentence can import another script or an action from action warehouse, "\QQ\QQlogin.pcap" is an action file which includes all real data packets about user login. The 'wait' and 'loop' sentences are used to control the interval among action and the number of loops respectively.

C. Test System

Fig. 3 shows the framework of the test system. A script compiler is implemented and deployed in TTCN (Testing and Test Control Notation) server to read scripts and generate test case files which can be understood by

machine. The replay software is implemented and deployed in application server and controller server to run test case files. The KQIs analysis tool is implemented and deployed in analysis server. The TTCN server, application server and analysis server are deployed in data center, and TTCN server remotely control the end-user equipments via application controller to run the test cases automatically.

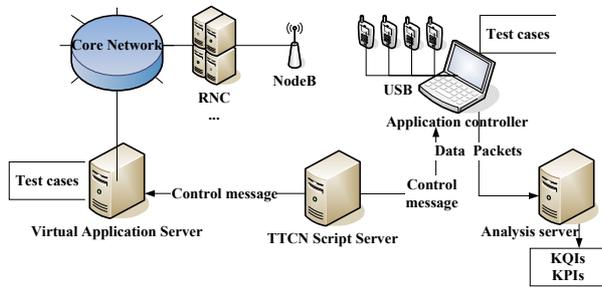


Figure 3. The framework of test system

The procedure of this system is given as follows.

- (1) Collect typical data packets according to user behavior model to build a warehouse.
- (2) Edit test case by using the script language mentioned above, and build scenario using TTCN script.
- (3) TTCN server extracts application layer data from data packets and assembles the instructions and application layer data into test case files.
- (4) TTCN server sends test case files, instruction messages and configuration information to application controller and server, and builds connection between controller and server.
- (5) Application controller and server replay test case on mobile network.
- (6) As the system begins to replay, the controller captures network data packets and sends them to KQIs analysis server.
- (7) KQIs analysis server computes the value of KQIs and reports results. Some KPIs will also be recorded for further analysis.

III. EXPERIMENT

Actually, a test system needs not to have the same results with the real commercial communication system. If a new scheduling strategy can improve the performance of certain applications in a test system, and after this strategy is deployed on commercial mobile network, the performance of those applications can get corresponding improvements, then the test system is effective.

But it is unrealistic to modify the scheduling strategy in commercial network only for proving the effectiveness of test system. So we design an experiment to examine the correlation between the test system and real applications. Firstly, both of the test system and real applications are connected to commercial network, and then we run the same test cases on them in very different scenarios. If the performance of the real application of scenario A is better than that of scenario B, and the performance of the test system in scenario A is also better than that of scenario B, then the test system and real applications are correlative, namely the test system is effective.

In this experiment, we select several scenarios. We use X and Y to denote the sets of average KQIs values of replay and real applications obtained in different scenarios respectively. We will observe PEARSON correlation between X and Y .

A. Applications selection and scenarios selection

The correlation test is based on the variation of application performance among various scenarios, but the difference might be offset by other factors. Thus there are two restrictions in our experiment. (1) The application server must work stably. (2) The variation of performance caused by difference among scenarios must be much larger than that caused by random fluctuation in the network.

QQ and SINA Micro Blog are respectively the most popular IM application and social network application in China. MSN is one of the most popular IM application in the world. The "sending message" action of QQ and MSN and "comment" action of SINA Micro Blog are the most frequent actions in these applications. And these actions only include a few data packets, so they are easy to analyze.

To survey the stability of the servers of the three applications, we test the delays of QQ sending messages, MSN sending messages, and comment in SINA Blog in an official building whose network has a high accessing speed. Fig. 4 describes the ratios between every delay and the average delay of each application.

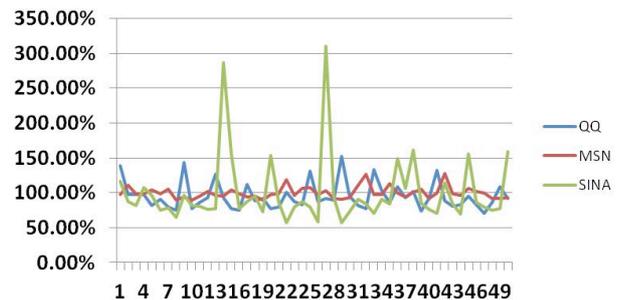


Figure 4. Delay fluctuation of QQ message, SINA Micro Blog comment and MSN message

The results show that the delays of QQ and MSN are obviously more stable than SINA Micro Blog. The "sending message" of QQ and MSN is also representative. QQ uses UDP protocol and the request/response mechanism works on the application layer, while MSN uses TCP protocol and the request/response mechanism works on the transport layer. So we choose QQ and MSN for the experiment.

Another key problem is how to select different scenarios according to condition (2). A scenario in commercial network should be composed by wireless channel environment factor, and traffic environment factor of data flow. According to [21-22], the mobile traffic model could be characterized by population, area (for example, city center, urban area, domestic or business zone) and time period (for example, rush hour, busy hour, and normal hour). According to this theory the common environment types can be divided into three categories: Indoor (such as domestic and business), outdoor (such as park and pavement), vehicle (such as bus, car and train). Finally, we choose two districts, an old district with higher

population density and a new district with lower population density, in Shanghai to conduct our experiment. According to the theory mentioned above, the scenarios include residential area in weekend daytime, downtown street in weekend daytime, shopping mall in weekend afternoon, office building in weekday work hours and moving car in weekday daytime.

B. Test and result analysis

In this experiment one person used QQ to send messages for about 100 times in 30 minutes. At the same time, the test system run "normal chat" and "frequent chat" test case to do the "sending message" action for 150 times in the same scenario. Both of them accessed to the same commercial network. For reducing the influence of random fluctuation, we delete the 5% largest delay from the records of replay business and real business.

In the MSN experiment we used the "normal chat" and "frequent chat" to activate the operation "sending message" for 200 times. In the moving car scenario, we only used "frequent chat" test case to activate 180 times. We also delete the 5% largest delay according to same reason.

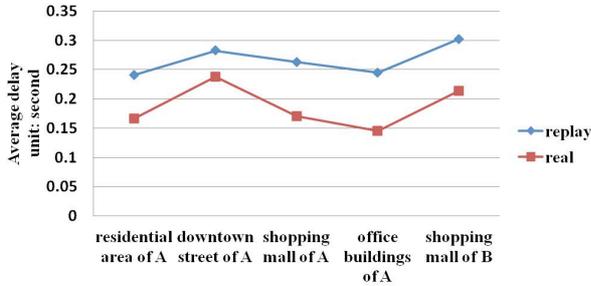


Figure 5. Average delay of QQ message in each scenario

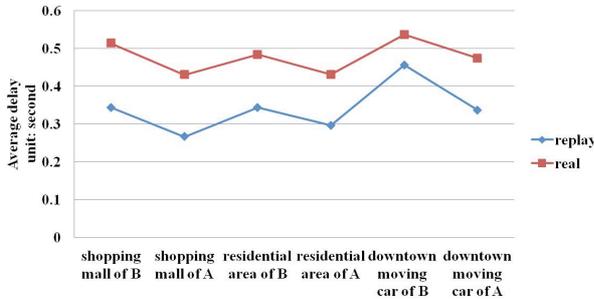


Figure 6. Average delay of MSN message in each scenario

Fig. 5 shows the average delay of QQ sending message of replay and real application in five different scenarios. Fig. 6 shows the average delay of MSN sending message of replay and real of MSN sending message in six different scenarios.

Let \bar{X}_i where \bar{X}_i is the average delay of replay application in the i th scenario. And let \bar{Y}_i where \bar{Y}_i is the average delay of real application in the i th scenario. The PEARSON correlation is:

$$\frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{Y})^2}} \quad (1)$$

The PEARSON correlation of QQ between replay and real applications is 84%, and that of MSN is 90%.

From this experiment, we conclude that the KQIs of replay applications is correlative with that of real applications, so this approach is effective.

IV. CONCLUSION

More new scheduling strategies begin to have the ability of assigning different schemes to different applications. To test this kind of new scheduling strategies, this paper proposes a new approach to evaluate performance of various applications from the viewpoint of QoE. A test system is implemented and experiment results prove the effectiveness of this system. Future works will focus on the similarity between the impact of simulation scenarios and corresponding real scenarios.

REFERENCES

- [1] H.J. Kushner, "Extensions of proportional-fair sharing algorithms for multi-access control of mobile communications: constraints and bursty data processes," in 2005 IEEE International Conference on Communications, Seoul, 2005, vol.5, pp.3149-3155.
- [2] C. Rosa and K.I. Pedersen, "Performance aspects of LTE uplink with variable load and bursty data traffic," in 2010 IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications, Istanbul, 2010, pp.1871-1875.
- [3] Hao Zhu and Guohong Cao, "On improving service differentiation under bursty data traffic in wireless networks," in IEEE INFOCOM 2004, Hong Kong, 2004, vol.2, pp.871-881.
- [4] I.V. Boichenko and E.V. Bortnikov, "Linux-based test-bed for testing of QoS subsystems in broadband wireless networks," in 2011 International Conference and Seminar of Young Specialists on Micro/Nanotechnologies and Electron Devices, Erlagol, Altai, 2011, <http://dx.doi.org/10.1109/EDM.2011.6006953>
- [5] Yonghuan Cao, et al., "The effect of access delay in capacity-on-demand access over a wireless link under bursty packet-switched data," Performance Evaluation, vol.57, pp.69-87, May. 2004.
- [6] Li Li, et al., "LTE CoS/QoS Harmonization Emulator," in 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, 2011, pp.154-161. <http://dx.doi.org/10.1109/CyberC.2011.34>
- [7] G. Piro, et al., "Simulating LTE Cellular Systems: An Open-Source Framework," IEEE Trans. Veh. Technol., vol. 60, pp.498-513, Feb. 2011. <http://dx.doi.org/10.1109/TVT.2010.2091660>
- [8] K.V. Vishwanath, and A. Vahdat, "Swing: Realistic and Responsive Network Traffic Generation," IEEE/ACM Trans. Networking, vol.17, pp.712-725, Jun. 2009. <http://dx.doi.org/10.1109/TNET.2009.2020830>
- [9] A. Klemm, et al., "Traffic modeling and characterization for UMTS networks," in 2001 IEEE Global Telecommunications Conference, San Antonio, Texas, 2001, vol.3, pp.1741-1746.
- [10] G. Judd, and P. Steenkiste, "A Simple Mechanism for Capturing and Replaying Wireless Channels," in Proc. of the ACM SIGCOMM 2005 Workshop on experimental approaches to wireless network design and analysis, Philadelphia, PA, 2005, <http://dx.doi.org/10.1145/1080148.1080161>
- [11] P. Chatterjee, et al., "Modeling the Clickstream: Implications for Web-Based Advertising Efforts," Marketing Science, vol.22, pp.520-541, May. 2003.

- [12] B. Fabricio, et al., "Characterizing User Behavior in Online Social Networks," in Proc. of the 9th ACM SIGCOMM conf. on Internet measurement, New York, NY, 2009, pp.49-62.
- [13] V. Bimal, et al., "On the evolution of user interaction in Facebook," in Proc. of the 2nd ACM workshop on Online social networks, New York, NY, 2009, pp.37-42.
- [14] W. Christo, et al., "User interactions in social networks and their implications," in Proc. of the 4th ACM European conference on Computer systems, New York, NY, 2009, pp.205-218.
- [15] *Estimating end-to-end performance in IP networks for data applications*, ITU-T Recommendation G.1030, 2006.
- [16] G. Frédéric and B. Sergio, "Towards real-time anomalies monitoring for QoE indicators," *annals of telecommunications - annales des télécommunications*, vol. 65, pp.59-71, Feb. 2010.
- [17] *GB923: Wireless Service Measurements Handbook*, 3rd ed., TM Forum, 2004.
- [18] *GB917-2: SLA Management Handbook*, 2.5 ed., TM Forum, 2005.
- [19] K. Dae-Woo, et al., "Analysis of Service Transaction Flow Based on User's Actions to Develop KQIs for WiBro Service," in Fourth Advanced International Conference on Telecommunications, Athens, 2008, pp.77-84.
- [20] S. Fabian, et al., "Understanding online social network usage from a network perspective," in Proc. of the 9th ACM SIGCOMM conf. on Internet measurement, New York, NY, 2009, pp.35-48.
- [21] J.G. Markoulidakis, et al., "Traffic model for third generation cellular mobile telecommunication systems," *Wireless Networks*, vol. 4, pp. 389-400, Sep. 1998. <http://dx.doi.org/10.1023/A:1019181028954>
- [22] J.G. Markoulidakis, et al., "Mobility modeling in third-generation mobile telecommunications systems," *Personal Communications*, vol.4, pp.41-56, Aug. 1997. <http://dx.doi.org/10.1109/98.612276>

AUTHORS

Lei Chen is with School of Electronics and Information, Tongji university, Shanghai, China (e-mail: 2010_dx_cl@tongji.edu.cn).

Ping Wang is with School of Electronics and Information, Tongji university, Shanghai, China (e-mail: shaffer2008@gmail.com).

Fuqiang Liu is with School of Electronics and Information, Tongji university, Shanghai, China (e-mail: fuqiangliu@163.com).

Chao Wang is with School of Electronics and Information, Tongji university, Shanghai, China (e-mail: chaowang7911@hotmail.com).

Haibo Zhou is with the Huawei Technologies Co. Ltd., Shenzhen, China (e-mail: zhouhaibo119@huawei.com).

Lijun Pu is with the Huawei Technologies Co. Ltd., Shenzhen, China (e-mail: pulijun@huawei.com).

Jiping Xiong is with the College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua, China (e-mail: xjping@ustc.edu).

Nguyen Ngoc Van is with the School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam (e-mail: ng_ng_van@yahoo.com).

This research was supported by the National Science and Technology Major Project of China under Grant 2012ZX03001033-005. Submitted 19 June 2013. Published as re-submitted by the authors 22 October 2013.