Performance Evaluations of Cloud Radio Access Networks

Mu-Han Huang, Yu-Cing Luo, Chen-Nien Mao, Bing-Liang Chen, Shih-Chun Huang, Jerry Chou, Shun-Ren Yang, Yeh-Ching Chung, and Cheng-Hsin Hsu(✉)

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
chsu@cs.nthu.edu.tw

Abstract. With the skyrocketing amount of data communications, traditional Radio Access Networks (RANs) infrastructure suffers from high capital and operating expenditures. Many countries and mobile network operators, therefore, propose software-defined radio access networks for centralized management, and further apply cloud computing technologies into cellular networks. Cloud Radio Access Network (Cloud-RAN) is a new paradigm for the next generation mobile network which provides ultra-high density deployments, dynamic reconfiguration of computing resources, as well as achieves high energy efficiency. To quantify the performance of Cloud-RAN infrastructure deployment, we build up real Software RAN testbeds based on an opensource LTE implementation over the latest virtualization technologies. We evaluate the performance of different testbed deployments by several test scenarios, in order to show the overhead introduced by virtualization. In addition, our testbed setup and measurement methodology will stimulate more systems research on the emerging Cloud-RAN infrastructure.

Keywords: OpenAirInterface · Virtualization · KVM · Docker · Container · RAN · Software RAN · Cloud RAN · Testbed

1 Introduction

The global mobile data traffic in 2015 was 55 % higher than 2014. Research predicts that the amount will raise 9 times as against 2014 while in 2020, and 80 % of mobile data traffic will be from smartphones by that time [4]. With the rapid developments of Machine-to-Machine (M2M) communications, large amount of data traffics impacts the current Radio Access Networks (RANs). To sustain tens of thousands of devices connected simultaneously, the next generation mobile network should achieve low latency and high throughput. However, the traditional RAN uses dedicated hardware for baseband processing which is lack of flexibility and scalability, and also leads to high Capital Expenditure (CAPEX) and Operation Expenditure (OPEX). Therefore, many countries and cellular network operators started to focus on Software RAN developments.

J.-H. Lee and S. Pack (Eds.): QShine 2016, LNICST 199, pp. 312–321, 2017.
DOI: 10.1007/978-3-319-60717-7_31

Compared to specialized hardware systems, the programmability, extensibility, and adaptability of commodity hardware turn Software RAN into one of the most promising solutions.

To be energy- and cost-efficient, recent studies propose to deploy Software RANs in cloud platforms [8,12], referred to as Cloud Radio Access Network (Cloud-RAN). In Cloud-RAN, baseband processing is centralized in a virtualized BaseBand Unit (BBU) pool. It allows the heterogeneous traffics to be handled by a share resource pool, and is able to adapt to different types of traffics. Moreover, researchers [19] propose Cloud RAN-as-a-Service concept, a new way to manage mobile networks. It not only improves the network throughput by centralized processing, but also takes advantages of cloud computing to increase the flexibility of resource usages. However, the existing cloud platforms are mostly developed for general purpose computing, and thus some concerns such as the network Quality of Service (QoS) and time-sensitive resource management mechanisms may not be rigorously studied and designed yet.

In this paper, we aim to evaluate the performance of Cloud-RAN in a realistic setup. We build a Software RAN testbed on top of physical machines, as well as in a container-based virtualization platform to discuss the possible overhead introduced by centralization. In our experiments, we use commercial User Equipments (UEs) to send different types of real network traffics. Through the profiling of computing resource usage, and the measurement of end-to-end network performance, we provide comprehensive evaluations over different platforms to discuss the critical issues of Cloud-RAN deployment. The rest of the paper is organized as follows. In Sect. 2, we introduce proposed Software RAN approaches, as well as some prior studies on Cloud-RANs. Section 3 shows our testbed architecture, including physical machines and containers. The performance evaluations over our testbeds are given in Sect. 4. We conclude the paper in Sect. 5.

2 Related Work

Many countries and cellular network operators have proposed Software RANs to provide a centralize-controlled, flexible, and evolvable architecture. As we introduced in our previous work [14], projects such as FluidNet [13,20], a Cloud-RAN prototype with a BBU pool can be adopted in various logical front-haul configurations. With FluidNet's algorithms, the traffic sustainability can be maximized to meet the real-time requirements, while simultaneously optimizing the system resource usage of BBU pool. Gudipati et al. [10] proposed a software-defined RAN with a centralized control plane. However, it only includes the control algorithm to make decisions over handover and interface management, no centralized baseband processing is done in the cloud. OpenRAN [23] is a Software RAN architecture that achieves the virtualization and programmability. With Software-Defined Networking (SDN), it has the capability to dynamically optimize the rules for each virtual access element. As for real-testbed that can be actually deployed, OpenBTS [17] is an open source cellular infrastructure that

allows users to deploy their own GSM network. However, it only supports 2G/3G networks. The aforementioned studies do not capitalize the characteristics of the cloud, nor quantify the performance of cloud-RAN over real 5G cellular network testbeds deploy in the cloud.

To move from Software RAN to Cloud-RAN, the balance between performance and expense is the most important issue, as well as to exhibit the characteristics of cloud, such as scalability, elasticity, and reliability. Pompili et al. [18] not only provided a comprehensive survey on Cloud-RAN, addressing its technical challenges and relevant open research issues, but also proposed resource provisioning and allocation strategies of BBU pooling. They also built a real-time testbed to compare the CPU and power consumption of a Cloud-RAN architecture against traditional approach to show the benefits of their solution [11]. To implement Cloud-RAN, the latency and real time issue should be carefully considered. In [15], Navid discussed critical issues on the RAN cloudification. Moreover, he proposed the splitting strategies of BBU and Remote Radio Head (RRH). Form his simulation results, he considered different scenarios, which affect the processing ability of BBU and model individual components of BBU functions. Different from the aforementioned studies, the current paper presents detailed performance evaluations using a real Cloud-RAN testbed.

3 The Considered Cloud RAN

3.1 Cloud System Architecture

Compared to conventional computing, cloud computing [7,9] makes more elastic use of computing resources without paying a premium for infrastructure deployment. It implies a service-oriented architecture that has better flexibility, scalability, and on-demand services. The Infrastructure-as-a-Service (IaaS) provider, such as Amazon's EC2 [1], provides the infrastructures for cloud consumers to build, run, and deploy their own services or platforms. Virtualization is extensively used in this case in order to abstract away and isolate the lower level functionalities. Kernel-based Virtual Machine (KVM) [5] is a widely-used full virtualization solution for Linux distributions. It turns the entire Linux kernel into a hypervisor, and completely simulates the underlying hardware including network card, disk, CPU, RAM, and etc. KVM is able to work with a great variety of guest OSs, and provides high isolation among users. However, full virtualization leads to longer launch time, and a complete network stack that requires extra network acceleration technologies such as hypervisor bypass to ensure high network performance.

The emerging container-based virtualization becomes more and more popular. Instead of running an entire kernel, containers only run as isolated processes in user namespace. That is, containers have considerable performance advantages in many aspects such as low network latency, near-native performance on memory, and almost identical computation speed [21,22]. Docker [2] is an opensource project that automates service deployment in containers. With its own libcontainer to access the virtualization features of Linux kernel, and the adoption of

layered file system (AUFS), Docker has become the state-of-the-art approach in lightweight virtualization technologies. Docker provides high-level APIs for users to build, ship, and run applications in containers, as well as image registry for developers. User can simply pull a pre-built image to launch an application in short time, or even pack their own instances as services and push them back to the repository.

3.2 From Soft-RAN to Cloud-RAN

In our previous work [14], we had deployed the OpenAirInterface (OAI) [16], an opensource Software RAN implementation, on commodity PCs and conducted several performance experiments in different virtualization environments. From the previous experiment results, we observed that fine-tuned real-time kernel significantly improves the network and computational latency regardless of virtualization techniques. Both Docker and virtual machines under system loads achieve 13.9 and 3.8 times improvement compared to generic kernel respectively. According to the results, Docker containers outperform virtual machines in both network and computational latencies.

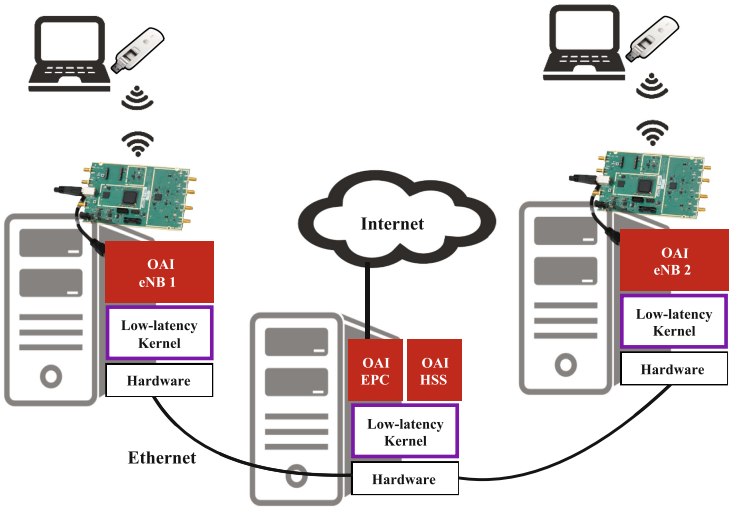
We proposed using real-time kernel and container to virtualize the Software RANs in the cloud. However, to further evolve from Software RAN to Cloud-RAN, we study following research problems in this paper: (1) how to deploy Software RAN in a virtualized environment, and (2) how to identify the performance bottleneck of cloud architectures for Cloud-RAN implementation. Furthermore, we use real mobile traffics to quantify the performance of our 5G cloud in the current paper, while we used general benchmark utilities in our earlier work [14].

4 Performance Evaluations

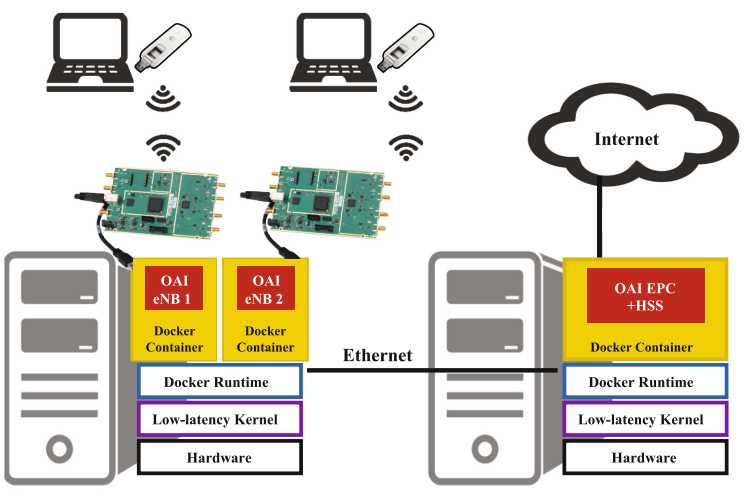
4.1 Testbed Design

In this paper, we aim to deploy OAI testbed on bare-metal machines and in containers for evaluating Cloud-RAN. Each physical machine comes with an AMD A10-7850K APU at 3.7 GHz with 4 CPU cores and 6 GB RAM. The OAI software is deployed on top of Ubuntu 14.04 with the low latency kernel 3.19. We turned off the power management features and maximize the CPU frequency for better performance and stability. We use National Instrument/Ettus USRP B210 as the RF front end, and Hauwei E3372 LTE dongle with a configurable SIM card as the UE, which connects to the Internet via the OAI software.

In order to focus on Evolved Node B (eNB) performance evaluations, we put Evolved Packet Core (EPC) and Home Subscriber Server (HSS) on the same entity (machine or container) to simplify the deployment. Figure 1(a) shows the bare-metal environment. The eNB and EPC+HSS are connected via Ethernet. eNB sends a connection setup request before attaching the UE to the eNB. After UE completes the RRC connection setup with eNB, the authentication between MME and HSS is accomplished, and the UE is able to access the Internet. On the



(a) Bare-metal environment.



(b) Container environment.

Fig. 1. Architecture of our OAI testbed.

other hand, the container testbed is shown in Fig. 1(b). We use Docker version 1.9.1 to achieve fast deployment of containers for eNB and EPC+HSS. Each container is able to utilize at most a CPU core, and at most 20% of memory by default. We consolidate eNB containers in one machine, while EPC+HSS in the other, connected by a Linux bridge.

4.2 Test Scenarios

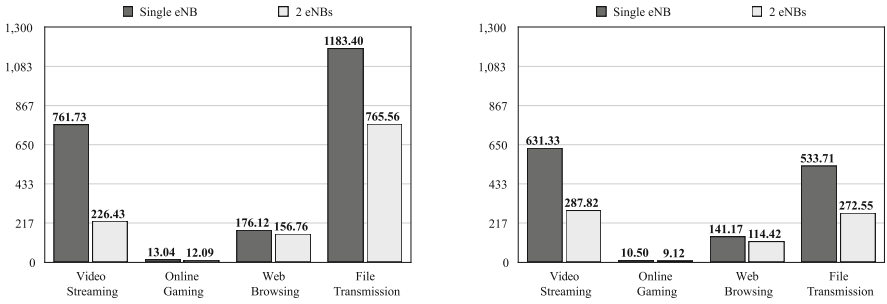
To evaluate the performance of our testbed, we generated 4 types of representative mobile traffics: (1) video streaming, (2) online gaming, (3) web browsing (social networking), and (4) file transmission. This is done by recording actual packets generated by each application using libpcap running on a 4G smartphone. Each packet record lasts for several minutes, and we rewind and replay them in our experiments once reaching their ends.

4.3 Evaluation Results

Network Throughput. In bare-metal environment, we measure the required bandwidth over four scenarios with a scale up to 2 eNBs. Figure 2(a) shows the comparison of 1 and 2 eNBs concurrently served by one EPC on a physical machine. Video-streaming requires about 761.726 KB/s throughput for users to have good user experience while watching a 1080p high-definition video. Online-gaming and Web-browsing use relatively low bandwidth at about 13.040 KB/s and 176.123 KB/s respectively. File-transmission is in high demand in throughput, we download a large tar file from the Internet and get an average throughput at about 1183.404 KB/s. With two eNBs, we observe that the available bandwidth is equally shared by the two eNBs.

For the container environment, Fig. 2(b) plots the comparison of 1 and 2 eNBs consolidated on one physical machine. Docker containers have comparable results in video-streaming, online-gaming and Web-browsing. However, it can only provide 553.706 KB/s on average for file-transmission. This can be partially attributed to the bursty nature and large data amount of file transmission, which impose higher consolidation burdens.

System Loading: CPU, Memory. We also profile the CPU and Memory usage to study the resource utilization of Cloud-RAN. When no traffic is incurred, as shown in Fig. 3(a), the idle Software RAN needs about 34.6% of CPU

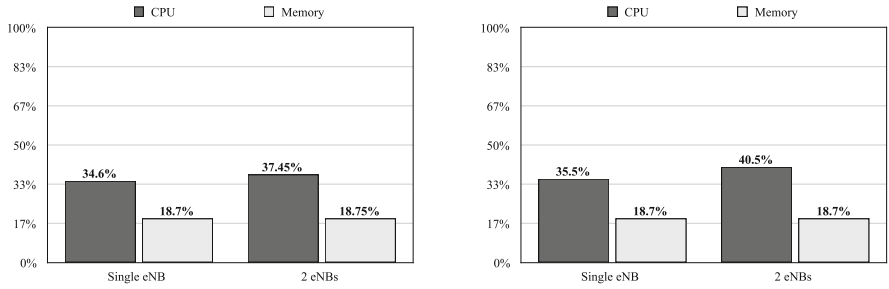


(a) Bare-metal environment. (b) Container environment.

Fig. 2. Network throughput performance in KB/s.

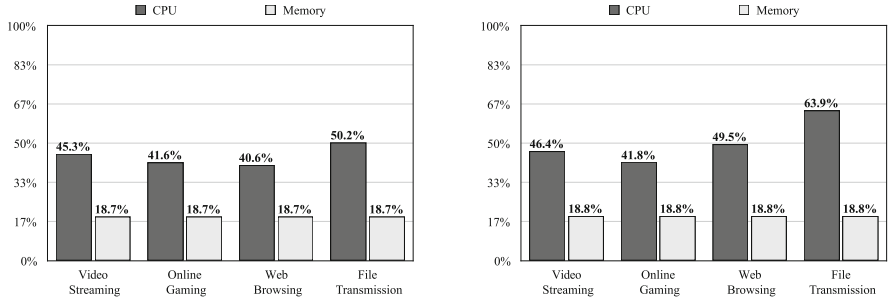
resources and 18.7% of memory. If we use a single EPC to serve 2 eNBs, each eNB requires 37.5% of CPU, but does not consume more memory. Figure 3(b) shows that using containers to deploy RAN service has slightly higher demand on CPU resources at about 35.5% for single eNB deployment, and 40.5% for 2 eNB serving at the same time. In bare-metal environment, likewise, we consider all four scenarios to study how different traffic types affect resource usage of eNB. In Fig. 4, we find that the CPU usage is affected by the used bandwidth. File-transmission scenario makes the highest utilization of CPU resources (50.2%), when other scenarios use about 42.5%. Almost the same observations are made, when we deploy 2 eNBs in our testbed, where the CPU usage is at most 1.3 times higher than that with a single eNB.

Figure 5 shows the performance results of the container environment, similar as the bare-metal environment, high throughput leads to high CPU usage. Video-streaming scenario, with the highest throughput of up to 631.326 KB/s, requires CPU usage of up to 63.1%. While 2 eNBs are deployed, 65.6% of CPU on average is used by each eNB. In summary, deploying Software RAN in container-based virtualization introduces at most 1.4 times of CPU loading compared to bare-metal deployment, and with little memory overhead.



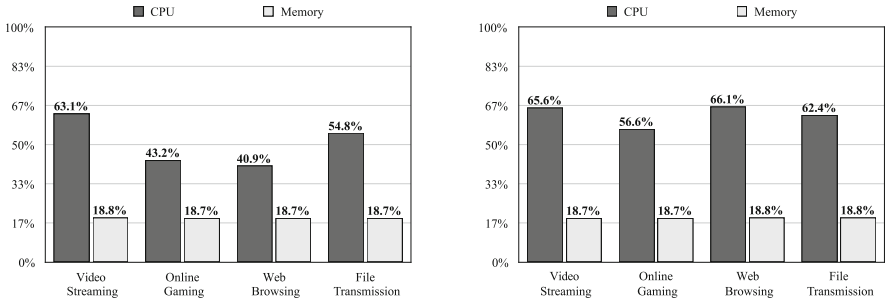
(a) Bare-metal environment. (b) Container environment.

Fig. 3. System resource usage at the idle time.



(a) Single-eNB deployment. (b) 2-eNB deployment.

Fig. 4. System performance in the bare-metal environment.



(a) Single-eNB deployment. (b) 2-eNB deployment.

Fig. 5. System performance in the container environment.

5 Conclusion and Future Work

To move Software RANs into the cloud, lightweight virtualization with high flexibility is indispensable. We deployed each Software RAN component in containers to mitigate the overhead introduced by virtualization. In the evaluation results, deploying Software RAN in the container cloud only increases at most 1.4 times of CPU loading compared to the bare-metal deployment and shows no negative impact on memory usage. Our next step is to launch several eNB services in containers to construct a resource pool of eNB services. We plan to use Kubernetes [6], an opensource cloud orchestration for the management of containerized applications in clustered environments. When network congestion occurs, we can easily deploy more eNB services to reduce the system loads. Moreover, the Replication Controller of Kubernetes can immediately recover from crashed services for higher overall stability. Our eventual goal is to deploy a complete 5G solution on a real-time container cloud platform with flexible deployment, dynamic resource allocation, and complete fault tolerance mechanism, which guarantee the overall performance of 5G networks. We will also leverage several technologies, such as Data Plane Development Kit (DPDK) [3] and SDN, for optimizing the 5G Cloud RAN solution.

Acknowledgment. This study was conducted under the Advanced Communication Technology Research and Laboratory Development project of the Institute for Information Industry, which is subsidized by the Ministry of Economic Affairs of the Republic of China. This work was also partially supported by Ministry of Science and Technology (MOST) of Taiwan under grant number MOST104-3115-E-007-004.

References

1. Amazon EC2. https://aws.amazon.com/ec2/
2. Docker web page. https://www.docker.com/
3. DPDK web page. http://dpdk.org/

4. Ericsson mobility report (2015). http://www.ericsson.com/res/docs/2015/ericsson-mobility-report-june-2015.pdf
5. Kernel-based Virtual Machine. http://www.linux-kvm.org/
6. Kubernetes. http://kubernetes.io
7. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
8. Checko, A., Christiansen, H.L., Yan, Y., Scolari, L., Kardaras, G., Berger, M.S., Dittmann, L.: Cloud RAN for mobile networks - a technology overview. Commun. Surv. Tutor. IEEE **17**(1), 405–426 (2015)
9. Dillon, T., Wu, C., Chang, E.: Cloud computing: issues and challenges. In: Proceedings of IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 27–33 (2010)
10. Gudipati, A., Perry, D., Li, L., Katti, S.: SoftRAN: software defined radio access network. In: Proceedings of ACM Workshop on Hot Topics in Software Defined Networking (HotSDN), pp. 25–30 (2013)
11. Hajisami, A., Tran, T.X., Pompili, D.: Dynamic provisioning for high energy efficiency and resource utilization in Cloud RANs. In: Proceedings of IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 471–472 (2015)
12. Lin, Y., Shao, L., Zhu, Z., Wang, Q., Sabhikhi, R.: Wireless network cloud: architecture and system requirements. IBM J. Res. Dev. **54**(1), 4:1–4:12 (2010)
13. Liu, C., Sundaresan, K., Jiang, M., Rangarajan, S., Chang, G.: The case for reconfigurable backhaul in cloud-RAN based small cell networks. In: Proceedings of IEEE INFOCOM, pp. 1124–1132 (2013)
14. Mao, C., Huang, M., Padhy, S., Wang, S., Chung, W., Chung, Y., Hsu, C.: Minimizing latency of real-time container cloud for software radio access networks. In: Proceedings of IEEE International Workshop of Quality of Service Assurance in the Cloud (QAC), pp. 611–616 (2015)
15. Navid, N.: Processing radio access network functions in the cloud: critical issues and modeling. In: Proceedings of ACM International Workshop on Mobile Cloud Computing and Services (MCS), pp. 36–43 (2015)
16. Nikaein, N., Marina, M., Manickam, S., Dawson, A., Knopp, R., Bonnet, C.: OpenAirInterface: a flexible platform for 5G research. ACM SIGCOMM Comput. Commun. Rev. **44**(5), 33–38 (2014)
17. Pace, P., Loscri, V.: OpenBTS: a step forward in the cognitive direction. In: Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN), pp. 1–6 (2012)
18. Pompili, D., Hajisami, A., Viswanathan, H.: Dynamic provisioning and allocation in cloud radio access networks (C-RANs). Ad Hoc Netw. **30**, 128–143 (2015)
19. Sabella, D., Rost, P., Sheng, Y., Pateromichelakis, E., Salim, U., Guitton-Ouhamou, P., Girolamo, M.D., Giuliani, G.: RAN as a service: challenges of designing a flexible RAN architecture in a cloud-based heterogeneous mobile network. In: Proceedings of IEEE Future Network and Mobile Summit (FutureNetworkSummit), pp. 1–8 (2013)
20. Sundaresan, K., Arslan, M., Singh, S., Rangarajan, S., Krishnamurthy, S.: FluidNet: a flexible cloud-based radio access network for small cells. In: Proceedings of ACM MobiCom, pp. 99–110 (2013)

21. Wes, F., Alexandre, F., Ram, R., Juan, R.: An updated performance comparison of virtual machines and Linux containers. In: Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 171–172 (2015)
22. Xavier, M., Neves, M., Rossi, F., Ferreto, T., Lange, T., Rose, C.: Performance evaluation of container-based virtualization for high performance computing environments. In: Proceedings of International Conference on Parallel, Distributed, and Network-Based Processing (PDP), pp. 233–240 (2013)
23. Yang, M., Li, Y., Jin, D., Su, L., Ma, S., Zeng, L.: OpenRAN: a software-defined ran architecture via virtualization. ACM SIGCOMM Comput. Commun. Rev. **43**(4), 549–550 (2013)