USD: A User-Centric Software Defined Platform for 5G Mobile Devices

Kien Nguyen(✉), Kentaro Ishizu, and Fumihide Kojima

National Institute of Information and Communications Technology,
3-4 Hikarinooka, Yokosuka, Kanagawa 239-084, Japan
{kienng,ishidu,f-kojima}@nict.go.jp

Abstract. This paper introduces USD, a novel User-centric Software Defined platform for 5G mobile devices, which supports a wide range of users with the diversity of technical experience. Respecting user preferences, USD is able to exploit multiple wireless networks, as well as, to differentiate application traffic. The advantages of USD are realized by using a set of network virtualization (NW) and Software Defined Networking (SDN) technologies. Similar to the state-of-the-art works, USD leverages SDN in the exploitation of multiple networks. However, USD uniquely uses network namespace to isolate an application traffic at a granularity as fine as a process (i.e., each process's traffic belong to one networking stack). Moreover, USD relaxes the dependence on radio hardware by using wireless virtualization. The relaxation aims not only to efficiently utilize 5G networking resources but also to add an user-centric interface. As a proof of concept, we implement a prototype of USD using the Wi-Fi, Open vSwitch, and the network virtualization technologies. We evaluate the performance of USD in a comparison with a legacy platform in an assuming 5G scenario. The evaluation results show that the USD prototype achieves comparable performances to the legacy platform while it introduces the advanced user-centric features.

Keywords: 5G · User-centric · Device · NW · SDN

1 Introduction

It is widely recognized that the functional 5G system will be available by the year 2020. Generally, the 5G system has been designed toward the technical key performance indicators (KPIs) such as peak data rate > 1 Gbps, end-to-end latency < 1 ms, etc. [1]. The 5G infrastructure is expected to provide extremely high bandwidth, low latency, reliable connection, which aims to significantly enhance the quality of user experience (i.e., the final goal). Therefore, among the most important issues, the 5G design is shifted from system-centric to user-centric [2]. That means the users will play an active role in networking functions in the 5G era. The users, who have different technical backgrounds, experience many evolving and novel applications (e.g., Big Data applications). The applications, each of which has different KPI requirements, should satisfy user expectation.

B. Mandler et al. (Eds.): IoT 360° 2015, Part II, LNICST 170, pp. 530–538, 2016.
DOI: 10.1007/978-3-319-47075-7_57

The 5G device hence should well support the traffic differentiation and user diversity.

The 5G device is commonly envisioned to have multiple wireless technologies, which operate under the dense deployments of multiple types of cells (e.g., small cell, micro cell, pico cell, etc.) [3]. The first important function on a 5G device is concurrent exploiting multiple wireless networks (same or different technologies) for applications. The benefit of exploitation is intuitive since a 5G application could aggregate different network's bandwidth or achieve seamless handover. On the other hand, many 5G wireless technologies promisingly provide sufficient 5G KPIs as shown in the standard committees' goals for releasing the 5G versions (e.g., Wi-Fi in [4]). Moreover, the switching cost between different technologies on device is not negligible (e.g., energy consumption, reconfiguration, etc.). Therefore, the 5G devices should avoid the switching as much as possible.

Addressing the aforementioned issues, this paper proposes a novel User-centric Software Defined platform (namely USD) for 5G mobile devices. USD leverages the advantages of network virtualization (NW) and software defined networking (SDN) technologies in the user-centric 5G design. Specifically, the NW in a form of wireless virtualization relaxes the dependent of hardware radio for the aims of enhancing resource utilization and creating an user-centric interface. The interface, which is SDN-based and open, allows the USD device to interact with a SDN control component through a well-defined API (i.e., OpenFlow). Hence, USD supports a wide range of users, who determines policies provided from a control component located on a device, from an operator, or a third party partner. Another advantage of USD is that it utilizes the NW in the form of network namespace in creating multiple networking stacks for application process isolation. As a result, a USD user is able to select networks for an application at the granularity as fine as a process. Similar to the state-of-the-art works, a SDN switch that attaches multiple radio interfaces is also used to exploit multiple networks in USD.

As a proof-of-concept, we build a prototype of USD with the Wi-Fi, Open vSwitch, and the network virtualization technologies. The wireless virtualization is adopted to create virtual Wi-Fi interfaces, which assumingly exploit the huge capability of 5G Wi-Fi links. Open vSwitch [5] attaches one or several wireless interfaces, each of which is programmed to direct the application flows to a different network. Open vSwitch could be programmed by a local controller or a remote controller (i.e., via the user-centric interface). Moreover, the USD prototype uses two network namespaces, which create independent networking stacks for determined application processes. We evaluate the USD prototype in an assuming 5G scenario in a comparison with a legacy platform (i.e., Ubuntu networking stack). The results show that the USD prototype effectively exploits nearby Wi-Fi networks, introduces the user-centric feature while it still achieves a comparable performance to the legacy one.

The rest of paper is organized as follow. In Sect. 2, we describe the design of USD. Then, Sect. 3 includes USD evaluation results. In Sect. 4, we present the related works. Finally, we conclude the paper in Sect. 5.

2 USD Design and Implementation

2.1 Problem Description

In the 5G era, the advanced technologies on both the 5G infrastructures and devices are capable of providing sufficient capacity in terms of bandwidth, latency, reliability. The 5G users will experience a plethora of applications, which are evolved from the existing applications and totally novel ones (e.g., Internet of Things (IoT) and Big Data applications). Generally, the 5G applications will have various requirements and generate a diversity of traffic types and traffic load. However, they should satisfy, or at least, avoid to degrade the quality of user experience. Many existing problems on the current legacy devices should be also solved in 5G. For example, network selection is following the user's criteria. That requires an important demand from the 5G user, that is controlling the application traffic with a fine granularity through user-expected networks.

On the other hand, the 5G network supports various types of users, who have different experience and technical levels. Therefore, the supported controlling level should be wide enough in order to support all of them. In this work, we consider an example of user classification, in which the 5G users are divided into expert, normal, novice categories. The expert user, who has good technical knowledge, should be able to build her/his own traffic control policies. The policy assigns a fine granularity of application to a network following her/his expectation. On the other end of controlling spectrum, the novice user with the limited technical experience simply relies on service providers (i.e., similar to in the current network). The normal 5G user, who has an average experience and technical knowledge, could define some simple policies his/herself. However, the normal user may refer for advanced policies from a third party (e.g., via a downloadable application). In order to support the users, it is highly required that a 5G mobile device should have an open interface to interact with the controlling components.

2.2 USD Design

In the user-centric software defined (USD) platform, we mainly leverage the power of software in controlling application traffic and radio access in a device (i.e., so-called software defined). The conceptual design of USD is presented in Fig. 1. In USD, an application process interacts with the module named *user preference*, which determines the process's destination networks. After assigning a user preference value, the process may directly follow either a private networking stack or an SDN-based one, which exploits one or several networks. In the case of using the private stack, the application works in a similar way as in the legacy one (i.e., Linux stack). Otherwise, it traverses through an SDN-based switch, which attaches one or several radios. The switch is controlled by a local or remote controller depending on the user expectation.

The advantage of using multiple networking stacks is that the service differentiation on the 5G device significantly improved. It allows the user to personalize

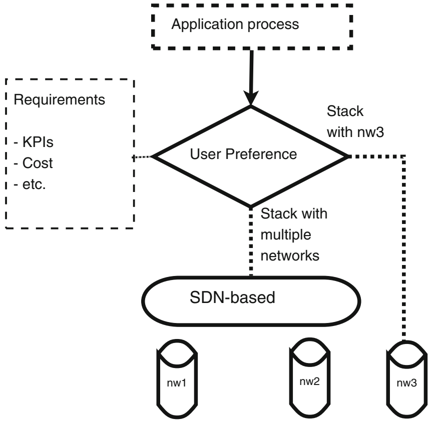


Fig. 1. User-centric Software Defined (USD) platform on a 5G device with two networking stacks and three networks (nw1, nw2, nw3)

the traffic flows with a fine granularity (i.e., the process level). For example, two identical traffic flows, which belong to different processes, could travel through different networks. Moreover, the stacks are created and controlled by software, hence they are efficient and flexible. In the SDN-based stack, USD also provides other SDN-based methods of traffic differentiation. The methods follow matching and forwarding based on various header fields of incoming packets (i.e., defined by the SDN/OpenFlow standard). The flexibility of SDN matching and forwarding is also inherited to handle the coexistence of multiple wireless networks. USD allows the SDN switch to associate both the virtual and physical wireless interface. The usage of virtual interface is two-fold: efficiently exploit the surrounding networks (e.g., several Wi-Fi APs) and to adding the user-centric interface. USD features the additional importance of the single networking stack since it is necessary in bypassing the limitations of the state-of-the-art SDN technology when the controller is remotely located from the devices. In the such scenario, the control traffic will be conveyed via that networking stack.

2.3 User-Centric Function

In USD, the user preference is one of the most important modules, which requires interactions from users. The module could be configured proactively or reactively depending the user experience. For example, the novice user may prefer to use the proactive configuration provided by an expert (i.e., operators or third party) hence the technical task is transparent to the user. On the other hand, the expert user may require a reactive action per one application process. In USD, the simplest form of the user preference module is a user's decision (e.g., from an expert user). However, it is commonly expected that the user is suggested the required KPIs of application. Moreover, the networks information such as bandwidth, cost, end-to-end delay, or another context-aware values are available for the user before making the decision. The technical and context-aware information could be provided either by several daemons running on the device or by suggestions from a controlling point in the networks.

2.4 Implementation Challenges

We implement USD on the open operating system Linux. One of the major challenges is isolating different application processes. We solve that challenge by the network namespace, which is one form of NW. When a network namespace is created, it includes a networking stack that could directly interacts with the application layer on a Linux host. The networking stack in each network namespace operates as a normal networking stack and provides sufficient functions for applications (such as routing tables, association, ARP caches, etc.). A different network namespace has a distinct set of kernel structures for networking, hence the two processes in two different network namespaces could be isolated. The network namespace is well-known for creating the widely-used emulator mininet [6]. In this work we introduce a novel use case in wireless environments.

Another challenge in USD is how to exploit multiple wireless networks efficiently. Besides the network namespace's isolation, the multiple networks should benefit the application on 5G devices. We follow the state-of-the-art approach in SDN, which bridges several wireless interfaces in a programmable switch. There are several options for the implementation of the switch such as Linux bridge, Open vSwitch, etc. However, we select Open vSwitch USD since it well supports design features. First, Open vSwitch not only can attach wireless interfaces but also provides open APIs for implementing controlling policies (i.e., OpenFlow). Second, the Open vSwitch's new versions have been proven to provide high performance. Note that, we don't attach all the wireless interfaces on a devices to Open vSwitch since it will isolate the switch with the network infrastructure. In order to interact with Open vSwitch, we use virtual Ethernet (veth) interfaces. The virtual Ethernets are special interfaces that always appear in pairs. They work like a tube, in which the traffic is in at one side and out in other side. We attach one side of the tube to the Open vSwitch while the other behaves as a normal network interface in the Linux networking stack.

3 Evaluation

This section presents evaluations of a USD prototype in an assuming 5G scenario, where the link capacity is sufficient. We built the USD prototype on Ubuntu 13.10, which includes Open vSwitch version 2.3.1, wireless virtualization, network namespace, and Wi-Fi technologies. The USD interfaces with the user preference module via a virtual Ethernet interface. Note that, the virtual Ethernet includes a pair of interfaces (i.e., veth0 and veth1). Moreover, the traffic, which is an input of veth0, will be output of veth1 and vice versa. To efficiently utilize the 5G wireless link capacity, the wireless virtualization creates the two virtual wireless interfaces (i.e., vwlan1 and vwlan2) on the Wi-Fi card. The card

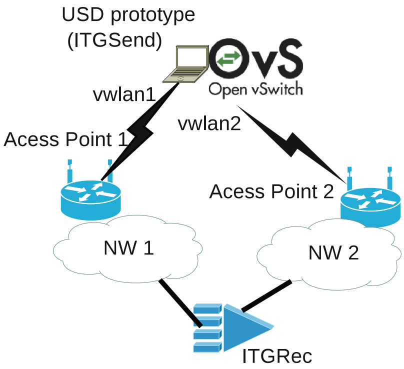
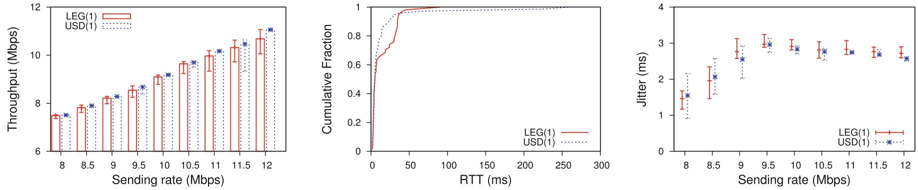


Fig. 2. Evaluation scenario

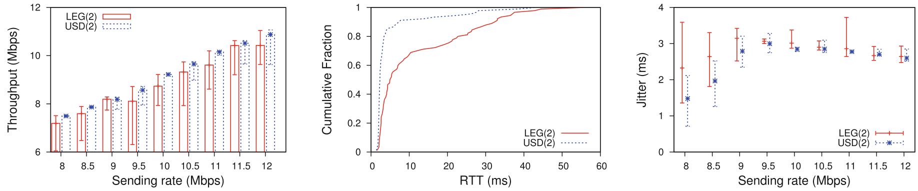
is Intel wireless card named Centrino Advanced-N 6250 (supporting IEEE 802.11 abgn) with the iwilwifi driver. We attach vwlan1 and veth0 to the Open vSwitch (OVS) with appropriate forwarding rules. In the evaluation, vwlan1 operates as one port of OVS, which routes the traffic via network one (NW1). vwlan2 is associated with the access point 2; then it forms a path to the network two (NW2). The connection diagram is shown in Fig. 2. The USD prototype is able to concurrently exploit the two Wi-Fi networks, to isolate application process, and to provide an user-centric interface. All the listed features are not supported on the legacy networking stack on Ubuntu. We then investigate the performance in order to evaluate the USD efficiency. In the evaluations, we use the simplest form of user preference; and compare the USD to the legacy networking stack (LEG). LEG includes the physical Wi-Fi card and the default network manager. The considered performance metrics include bandwidth, round trip time (RTT), and jitter. We use the distributed Internet traffic generator (DITG) [7] in this work. DITG is the traffic generator that accurately provides many traffic patterns such as the inter departure time between packets (IDT) and the packet size (PS) stochastic processes. In the operation, the ITGSend module of DITG is activated on the traffic generation machine, while the ITGRecv module is listening on the destination of the packets.

In the first evaluation, the ITGSend on the device sends UDP flows to a ITGRecv server (i.e., another Ubuntu 13.10). We assume that the user selects NW1 for the traffic flows. As a result, the UDP flows follows a network namespace's communication stack. We vary the sending rate from 8 Mbps to 12 Mbps with the step of 0.5 Mbps. With each value of sending rate, the constant rate remains in a period of 30 s. Moreover, each experiment at a rate step is repeated 10 times. The same process is experimented with LEG. We collect all the values of throughput, round trip time (RTT), and jitter and show them in Fig. 3. Note that, the RTT option of DITG is used to avoid the errors caused by clock synchronization between the machines of ITGSend and ITGRecv. In Fig. 3a, we plot the average, max, min values of throughput. We can observe



(a) Throughput (b) Delay (c) Jitter

Fig. 3. Performance comparison between USD and LEG using NW1



(a) Throughput (b) Delay (c) Jitter

Fig. 4. Performance comparison between USD and LEG using NW2

that similar to LEG, USD intuitively increases the throughput when the rate increases. However, the USD throughput is slightly better than the other. In Fig. 3b, the cumulative distribution functions (CDFs) of all the RTT values are plotted. Most of the values of USD's RTT are better than the ones of LEG. However, there are several exceptional. The same observation could be drawn from the jitter value's figure (i.e., Fig. 3c). Therefore, we conclude that USD and LEG has the comparable performances in the case of using network one. In the second evaluation, we aim to use NW2. The USD machine is configured to send the UDP flows through the virtual Ethernet interface. The flows will be processed by the Open vSwitch, which is necessarily preconfigured a set of appropriate rules. In this work, the rules are local but they could be installed by a remote controller. We also compare the performance of USD with LEG using NW2. We plot the performance values of throughput, RTT, and jitter in Fig. 4a, b, and c, respectively. Even getting additional processing steps, USD still keeps comparable performances in all metrics with LEG.

4 Related Work

The researchers have thoroughly investigated potential technologies for 5G systems such as mmWave, massive MIMO, full-duplex communication, SDN, cloud RAN, [8], etc. The recent results show that the technologies promisingly reach the PKIs in the next few years. Since the major final goal of 5G is satisfying the user expectation, the 5G design should be shifted from system-centric (as in the existing networks) to user-centric. In the scope of this work, we focus on the user-centric issue on the 5G mobile device. We begin with the common agreement of the 5G device, in which there is a coexistence of multiple wireless on a mobile device [9]. The user-centric 5G device should effectively exploit its surrounding wireless networks for the quality of user experience.

In the mobile network research, there are several works related to exploiting multiple networks. In [10], the Delphi framework lets the application interface with the transport layer, which is specially designed for network selection (i.e., based on predetermined policies). The Delphi design is application-driven, which is different to the user-driven approach in our work. However, we can inherit the Delphi's advanced features to enrich the USD platform. Another approach is applying multipath transport control protocol in wireless networks [11,12]. However, it also shares the listed characteristic with Delphi. Regard using SDN for user-centric features, the work in [13] provides an interesting analysis although it is lack of implementations or evaluations. The real SDN-based implementation [14], in which the SDN technology is used to provide flexible ways for conveying application flows, inspires us to design USD. In that work, the rewriting header technique (i.e., address translation) is used to stitch bandwidths or to achieve handover between different networks. However, the technique causes overhead (e.g., lowering throughput); and its operation has to rely on a local controller. In our USD design, we inherit the method of exploiting multiple networks from [14]. However, USD can avoid the complicated network translation mechanisms (using single stack). Besides that, USD introduces an open, user-centric interface, which supports a wide range of user and controlling locations. Additionally, USD is able to route application flows at the process granularity.

5 Conclusion

This paper introduces the User-centric Software Defined (USD) platform, which is designed towards the service personalization demand of 5G users. USD relies on several software-based technologies (i.e., SDN and NW) in order to provide a rich set of networking policy. USD supports a wide range of users, who could determine the network policies generated by themselves, network providers, or even third party vendors. Additionally, USD is able to differentiate the application traffic at a process level, as well as, to concurrently exploit multiple wireless networks. To show the effectiveness of USD, we first build a USD prototype using the Open vSwitch, network virtualization, network namespace, and Wi-Fi technologies. We then evaluate the USD prototype in an assuming 5G scenario in a comparison with the Linux's legacy networking stack. The evaluation results show that USD achieves comparable performances with the legacy stack including the physical Wi-Fi card. However, USD introduces the user-centric and several advanced features, which could not be supported on the legacy stack.

Acknowledgement. This work was supported by JSPS KAKENHI Grant Number 26730055.

References

1. Monserrat, J.F., Mange, G., Braun, V., Tullberg, H.M., Zimmermann, G., Bulakci, O.: METIS research advances towards the 5G mobile and wireless system definition. EURASIP J. Wirel. Commun. Netw. **2015**, 53 (2015)
2. Liotou, E., Elshaer, H., Schatz, R., Irmer, R., Dohler, M., Passas, N., Merakos, L.: Shaping QoE in the 5G ecosystem. In: Proceedings of IEEE QoMEX 2015, pp. 1–6 (2015)
3. Bhushan, N., Li, J., Malladi, D., Gilmore, R., Brenner, D., Damnjanovic, A., Sukhavasi, R., Patel, C., Geirhofer, S.: Network densification the dominant theme for wireless evolution into 5G. IEEE Commun. Mag. **52**(2), 82–89 (2014)
4. Rayment, S.: The Role of Wi-Fi In 5G networks. http://johannesbergsummit.com/wp-content/uploads/sites/6/2013/11/Rayment-Johannesberg-Ericsson-Rayment-pA7.pdf
5. Open vSwitch. http://openvswitch.org
6. Handigol, N., Heller, B., Jeyakumar, V., Lantz, B., McKeown, N.: Reproducible network experiments using container-based emulation. In: Proceedings of ACM CoNEXT (2012)
7. Botta, A., Dainotti, A., Pescapè, A.: A tool for the generation of realistic network workload for emerging networking scenarios. Comput. Netw. **56**(15), 3531–3547 (2012)
8. Boccardi, F., Heath, R.W., Lozano, A., Marzetta, T.L., Popovski, P.: Five disruptive technology directions for 5G. IEEE Comm. Mag. **52**(2), 74–80 (2014)
9. Bangerter, B., Talwar, S., Arefi, R., Stewart, K.: Networks and devices for the 5G era. IEEE Comm. Mag. **52**(2), 90–96 (2014)
10. Deng, S., Sivaraman, A., Balakrishnan, H.: All your network are belong to us: a transport framework for mobile network selection. In: Proceedings of ACM HotMobile 2014 (2014)
11. Raiciu, C., Paasch, C., Barr, S., Ford, A., Honda, M., Duchene, F., Bonaventure, O., Handley, M.: How hard can it be? Designing and implementing a deployable multipath TCP. In: Proceedings of USENIX NSDI (2012)
12. Nguyen, K., Ji, Y., Yamada, S.: A cross-layer approach for improving WiFi performance. In: Proceedings of IEEE IWCMC 2014, pp. 458–463 (2014)
13. Nunes, B.A.A., Santos, M.A.S., de Oliveira, B.T., Margi, C.B., Obraczka, K., Turletti, T.: Software-defined-networking-enabled capacity sharing in user-centric networks. IEEE Commun. Mag. **52**(9), 28–36 (2014)
14. Yap, K.-K., Huang, T.-Y., Kobayashi, M., Yiakoumis, Y., McKeown, N., Katti, S., Parulkar, G., Making use of all the networks around us: a case study in android. In: Proceedings of ACM SIGCOMM Cell Net, pp. 19–24 (2012)