Deep Learning-based Application Specific RAN Slicing for Mobile Networks

Ping Du
The University of Tokyo
Tokyo, Japan
Email: duping@iii.u-tokyo.ac.jp

Akihiro Nakao
The University of Tokyo
Tokyo, Japan
Email: nakao@nakao-lab.org

***Abstract*—Effectively identifying application is desirable for network operators to improve spectrum efficiency and user experience in future mobile networks that are expected to support multiple kinds of applications with different quality of service (QoS) requirements. In this paper, we present a Radio Access Network (RAN) slicing architecture utilizing in-network deep learning to apply application specific radio spectrum scheduling. We use a small number of customized supervising phones to generate training data in real-time and apply deep learning at the packet gateway (P-GW), where we tag the downlink packets with the identified application name and transmit them to eNB for application specific spectrum scheduling. The preliminary experimental results show the feasibility and the efficiency of the proposed application specific RAN slicing.**

***Keywords*—Software-defined Networking (SDN); Application Specific Optimization; Mobile Network**

I. Introduction

Although 5G Mobile networks will provide much more mobile bandwidth than current LTE networks, the area of radio resource allocation optimization will continue receiving significant interest as the operators face an increasing demand for mobile data traffic with various quality of service (QoS) requirements. Network slicing has been considered as one of the most significant technologies for 5G mobile networks [1], where mobile network operators can apply different spectrum allocation policies to different Radio Spectrum Network (RAN) slices to improve spectrum efficiency and user experience.

However, how to effectively identify and classify applications in real-time is still an open issue especially in the RAN research area due to the two main challenges. First, as long as data has been transmitted from user equipments (UEs) into a mobile network, the contextual information of the data (e.g., which application the data belongs to) is hidden from network alliances. Second, data packets will be compressed, concatenated and modulated in a RAN area, which makes application identification in a RAN much more difficult than that in a core network (CN).

To address this issue, we present an application specific RAN slicing architecture utilizing in-network deep learning. In our design, we apply in-network deep learning at Packet Gateway (P-GW) to identify mobile applications and attach the identification results to the downlink packets and transmit them to eNodeB (eNB). The eNB can apply application specific spectrum allocation policy based on the attached application information.

The main advantage of the proposed architecture is that we propose to apply deep learning-based application identification at packet gateway (P-GW). P-GW is the best point to perform application identification since all traffic needs to go through P-GWs. As a comparison, there could be only a very limited number of UEs connected to a single eNB so that we may not be able to collect enough training data at a single eNB.

Second, with applying application identification only once, we can apply the identification results in both RAN and CN. Where, we deploy application specific spectrum allocation at eNB using the identification results piggyback on the packets from P-GW to RAN. Meanwhile, we classify the uplink packets to different virtual network functions for application specific in-network processing [2].

Finally, we show the feasibility of our proposal with a prototype on our existing platform [1]. As far as we know, there is very few real implementation of application specific spectrum scheduling although there are many simulation-based works [3], [4] on how to schedule spectrum resources with different QoS requirements (e.g., real-time and delay-tolerant).

II. System Design

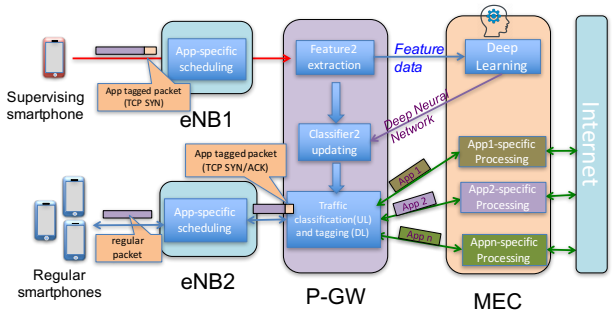

Fig. 1. Architecture of Deep Learning-based Application Specific RAN Slicing

As shown in Figure 1, we propose an architecture for application specific design of spectrum scheduling in RAN and packet processing in Mobile Edge Computing (MEC) in real-time utilizing deep learning on reliable training data

978-1-5386-6831-3/18/$31.00 ©2018 IEEE

generated by supervising smartphones. We use a small number of customized smartphones as supervising smartphones to generate training data where packets are tagged with the information of the application transmitting them. Then we apply deep learning on given flow and extract the useful features in a train of packets contained in the flow, without looking into the payload of packets. After identify application at the Packet Gateway (P-GW), we can (1) classify the uplink packets from UE to different virtual network functions in MEC for application specific in-network processing and (2) tag the downlink packets from MEC with the identified application name and transmit them to eNB. Then the eNB can apply application specific spectrum scheduling based on the attached application information. Since we have already shown the efficiency of application specific in-network processing in our previous work [2], we will focus on application specific spectrum scheduling in RAN in this paper.

A. Application Identification at P-GW

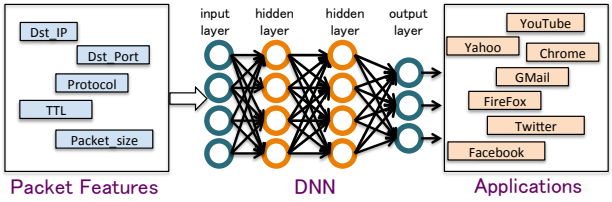

Fig. 2. Application Identification with Deep Neural Networks with Extracted Features

Conventionally, there are several ways to achieve application identification and classification, e.g., packet header marking [5], and deep packet inspection (DPI) [6] to identify application from packet payload. But packet header marking fails to identify a broad scope of applications while DPI is becoming harder and harder due to that application specific information conveyed in payload is mostly likely encrypted.

As shown in Fig. 2, our training model is defined based on deep neutral networks (DNN) with an input layer, multiple fully connected hidden layers and an output layer. Each layer is a feed-forward neural network.

To protect users' privacy, we use <dst_ip, dst_port, protocol, ttl, packet_size> as flow features without looking into the payload of packets. The detail of our DNN module could be found in our previous work [7].

B. Application Specific Spectrum allocation in RAN

The radio resource of LTE link can be divided in both time and frequency dimensions. The frequency dimension is divided into subcarriers. The time dimension is first divided into 10ms radio frames and each frame is further subdivided into ten 1ms subframes consisted of two 0.5ms slots. Transmission bandwidth depends on the number of active Resource Blocks (RBs) in a transmission. Each resource block can be allocated to only one user at a time slot.

Mobile operators are interested in improving their RB allocation algorithm to improve spectrum efficiency and user experience. For example, real time applications such as VoIP and video streaming have strict latency and throughput requirements while the traffic of some other applications (e.g., Email) can adjust to wide range of changes in delay and throughput and still meet the user expectations. In [3], [4], the rate allocation algorithm gives priority to real-time applications over delay-tolerant applications when allocating resources as the utility proportional fairness rate allocation policy is used.

Some others work on RAN slicing to provide different levels of radio resource isolation. FlexRAN [8] enables RAN slicing by decoupling the control from the data plane of base stations using a custom-tailored southbound API, where the radio resources need to be allocated among the connected UEs based on the requirements of the slice they belong to. The Orion [9] approach groups PRBs into vRB through a set of abstractions, and provides only relevant resource information to the corresponding slice.

None of the above works have addressed how to identify application and classify traffic to RAN slice. In most of existing work, mobile users are categorized based on applications running on their devices. In the proposed architecture shown in Fig. 1, we can reuse the application identification results at P-GW and piggyback the application information on the very first packets (e.g., TCP SYN/ACK) of each flow. There are two benefits of our design: (1) we don't need to apply deep learning based application identification on eNB so that the processing load of eNB could be reduced; (2) the traffic of all UEs must pass through the P-GWs while there is only a very limited number of UEs connected a single eNB so that we may not be able to get enough training data if we apply deep learning on the traffic collected at a single eNB.

C. Prototype of Application Specific RAN

We prototype the proposed application specific RAN slicing based on our previous work [1] with FLARE nodes [10] and OpenAirInterface (OAI) [11].

For the purpose of downlink spectrum scheduling, each UE probes the channel quality and reports the Channel Quality Indicator (CQI) to its associated eNB. The MAC layer of eNB then decides on the modulation scheme that can be scheduled to the UE and then check the physical resource grid for availability of the RBs. From this step the MAC can decide upon the modulation and coding scheme index (I_{MCS}) and then decide upon the number of resource blocks (RBs), which can be allocated to the UE. After this step, the maximum amount of data that can fit into a RB, named transport block size (TBS) is derived from the look up table as specified in the LTE phy specification [12].

The architecture of the OAI scheduling algorithm at MAC scheduler is as following: (1) calculates the average number of RBs for each active UE; (2) assigns the minimum RBs between requested RBs and average RB to each UE; (3) sorts active UEs according to some criteria such as the channel quality; (4) allocates the remaining RBs to high priority UEs according to the sorted list until no more RBs are available.

We first retrieve an IP packet encapsulated in GTP-U at the S1-U interface of an eNB. Then we check whether its

trailer is tagged with application name. In our prototype, only TCP SYN/ACK packets with zero-payload are tagged with application information. The application information of the following packets from the existing flows can be get from the <FlowID, App> table. The MAC layer will lookup the <App, Policy> table to schedule RBs according to the Policy. For a packet cannot be found in the <FlowID, App> table, the MAC layer will apply default RB scheduling policy.

One challenge is that the eNB schedules subframes at granularity of UE while we classify data into application at granularity of flow, where data from multiple flows of the same UE are assembled into one subframe at MAC layer. According to our previous research in [2], about 55% of UE devices launch only one TCP connection, and 70% of UE devices launch less than 2 concurrent TCP connections. So here we can safely assume that there is only main application running on each UE in a LTE network. In future 5G network, when a UE is running multiple applications with different QoS requirement, we need to assemble flows from different applications into different subframes even they are from the same UE.

III. System Evaluation

A. Application Identification at P-GW

We use two-week MVNO data as training data, where each day of traffic consists of about 40000 flows. Besides training and test, we use one-day of traffic as validation in training.

By using a tuple of <dst_ip, dst_port, protocol, ttl, packet_size> as the features of application traffic captured at an MVNO, we can successfully identify 200 mobile applications with about 93.5% accuracy over 39-day traffic using an 8-layer Deep Neural Network with TensorFlow [13], where each hidden layer consists of 40000 (200x200) neurons. The detail of the evaluation is shown in [7].

B. Application Specific Spectrum Scheduling in RAN

As we introduced in Sect. II, a naive OAI RB scheduling is done in two rounds. At the first round, we assign a maximum number of average number of RBs to each UE. The remaining RBs are allocated to high priority UEs. In our preliminary evaluation, besides the default two-round fairly scheduling policy, we also define a new greedy scheduling policy, where a UE running a high priority application can be assigned with requested RBs at the first round.

In our evaluation, we first check whether two UEs with default fairly-sharing policy can get bandwidth fairly. We run iperf receiver on both UEs and send UDP traffic to them from eNB. Fig. 3(a) shows that two UEs can get almost the same bandwidth when both applied the fairly-sharing policies (Test1). As a comparison, UE1 can get more bandwidth if we apply greedy policy to it in Test2.

Fig. 3(b) shows the experimental throughput results under different scheduling policies with two TCP applications: iperf and SpeedTest. The iperf traffic is scheduled with the default fairly-sharing policy while SpeedTest is scheduled with fairly-sharing and greedy policy in different tests. We

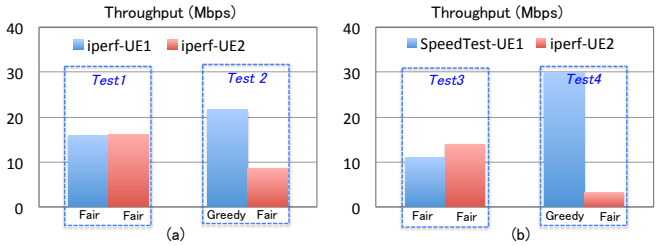

Fig. 3. Comparison of Greedy and Fairly-Sharing Scheduling Policies.

observe that the two UEs can share bandwidth fairly under default fairly-sharing policy (Test3). As a comparison, if we apply SpeedTest the greedy policy, the SpeedTest-UE1 can get more bandwidth than iperf-UE2 in Test4.

IV. Conclusion and Future Work

In this paper, we present an application specific mobile network architecture utilizing in-network deep learning to apply application specific radio spectrum scheduling in RAN. We show the efficiency of the proposal with prototype. Our future work will focus on user-defined RB scheduling policies.

Acknowledgments

This work has been partly supported by the EU-Japan coordinated R&D project 5G!Pagoda, and JSPS KAKENHI Grant number JP18K11255 in Japan.

References

[1] Akihiro Nakao and Ping Du et. al, "End-to-end network slicing for 5g mobile networks," *IPSJ Journal of Information Processing*, 2017.
[2] Ping Du and Akihiro Nakao, "Application specific mobile edge computing through network softwarization," in *Cloud Networking (Cloudnet), 2016 5th IEEE International Conference on*. IEEE, 2016, pp. 130–135.
[3] Tugba Erpek, Ahmed Abdelhadi, and T Charles Clancy, "An optimal application-aware resource block scheduling in lte," in *Computing, Networking and Communications (ICNC), 2015 International Conference on*. IEEE, 2015, pp. 275–279.
[4] Jun He and Wei Song, "Appran: Application-oriented radio access network sharing in mobile networks," in *Communications (ICC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3788–3794.
[5] Arthur Callado, Carlos Kamienski, Géza Szabó, Balázs Péter Gero, Judith Kelner, Stênio Fernandes, and Djamel Sadok, "A survey on internet traffic identification," *IEEE communications surveys & tutorials*, vol. 11, no. 3, 2009.
[6] Michelle Cotton, Lars Eggert, Joe Touch, Magnus Westerlund, and Stuart Cheshire, "Internet assigned numbers authority (iana) procedures for the management of the service name and transport protocol port number registry," Tech. Rep., 2011.
[7] Akihiro NAKAO and Ping DU, "Toward in-network deep machine learning for identifying mobile applications and enabling application specific network slicing," *IEICE Transactions on Communications E101-B*, vol. 14, pp. 153–163, 2018.
[8] Xenofon Foukas, Navid Nikaein, Mohamed M Kassem, Mahesh K Marina, and Kimon Kontovasilis, "Flexran: A flexible and programmable platform for software-defined radio access networks," in *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. ACM, 2016, pp. 427–441.
[9] Xenofon Foukas, Mahesh K Marina, and Kimon Kontovasilis, "Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 2017, pp. 127–140.
[10] "Flare: Open deeply programmable network node architecture," http://netseminar.stanford.edu/seminars/10_18_12.pdf.
[11] Navid Nikaein, Mahesh K Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet, "Openairinterface: A flexible platform for 5g research," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 33–38, 2014.
[12] ETSI Lte, "Evolved universal terrestrial radio access (e-utra); base station (bs) radio transmission and reception (3gpp ts 36.104 version 8.6. 0 release 8), july 2009," *ETSI TS*, vol. 136, no. 104, pp. V8, 2009.
[13] "Tensorflow," https://www.tensorflow.org/.