Demo: On enabling 5G Dynamic TDD by leveraging Deep Reinforcement Learning and O-RAN

Karim Boutiba
EURECOM
Biot, France
karim.boutiba@eurecom.fr

Miloud Bagaa
CSC-IT Center for Science Ltd.
Espoo, Finland
miloud.bagaa@csc.fi

Adlen Ksentini
EURECOM
Biot, France
adlen.ksentini@eurecom.fr

ABSTRACT

Dynamic Time Duplex Division (D-TDD) is a promising solution to accommodate the new emerging 5G and 6G services characterised by the asymmetric and dynamic Uplink (UL) and Downlink (DL) traffic demands. D-TDD dynamically changes the TDD configuration of a cell without interrupting users' connectivity, hence balancing the bandwidth for UL or DL communication according to the traffic pattern. However, 3GPP standard does not specify algorithms or solutions to derive the TDD configuration, i.e., the number of slots to dedicate to UL and DL. In [1], we have proposed a Machine Learning (ML)-based solution relaying on Deep Reinforcement Learning (DRL) to allow the base station (or gNB) to self-adapt to the traffic pattern of the cell by periodically adapting the number of slots dedicated to UL and DL. In this work, we implemented the DRL algorithm on top of an open-source gNB based on OpenAirInterface (OAI) [3] to demonstrate its efficiency. To this end, we relied on the O-RAN architecture [5], where the proposed DRL algorithm is deployed as xApp at the Near Real-time RAN Intelligent Controller (RIC) and communicates with the base station using O-RAN E2 interface. We developed xTDD Service Model (SM) following the E2SM standard [5], allowing the DRL solution to monitor DL and UL buffers from the gNB to deduce the optimal TDD configuration that accommodates the current traffic. Then, the decision (i.e., TDD configuration) is pushed to the base station. We implemented the solution on top of the OAI 5G StandAlone (SA) platform and Open Networking Foundation (ONF) RIC [4] based on μonos. To the best of our knowledge, this is the first demonstration of a ML-based D-TDD on top of a real 5G network, showing the advantage of O-RAN architecture to building Self Organized Network (SON) function for dynamic configuration of D-TDD.

CCS CONCEPTS

• **Networks** → **Network experimentation**; **Programmable networks**; **Network control algorithms**.

KEYWORDS

Dynamic TDD, Reinforcement Learning, OpenAirInterface, 5G, O-RAN

MobiHoc '22, October 17–20, 2022, Seoul, Republic of Korea

ACM ISBN 978-1-4503-9165-8/22/10.
https://doi.org/10.1145/3492866.3561252

ACM Reference Format:
Karim Boutiba, Miloud Bagaa, and Adlen Ksentini. 2022. Demo: On enabling 5G Dynamic TDD by leveraging Deep Reinforcement Learning and O-RAN. In *The Twenty-third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '22), October 17–20, 2022, Seoul, Republic of Korea.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3492866.3561252

1 INTRODUCTION

In recent years, the small cell deployment scenario has been experiencing increasing growth, especially for industry 4.0 use cases and 5G private networks. While 4G networks were designed to accommodate DL dominant traffic, 5G networks need to adapt to the new emerging services such as high-quality video streaming captured by drones for building surveillance that requires more UL traffic than DL. In this context, Dynamic TDD (D-TDD) is a promising solution introduced in 5G to satisfy the dynamic traffic pattern of small cell deployment. In D-TDD the allocation of the DL/UL ratio (i.e., the number of DL and UL slots in a frame) can be dynamically adjusted according to the UL and DL traffic demands. D-TDD allows the base station to change the TDD pattern dynamically without interrupting users' connectivity. Introduced flexibility by D-TDD allows the base station to adapt the frame configuration according to the traffic pattern by selecting the number of slots dedicated to UL and DL. However, the 5G NR specifications only cover the mechanism allowing the base station to inform the UE about the UL/DL slots pattern in a TDD frame, leaving the algorithm deriving the pattern UL/DL opens. In [1], we have filled this gap by proposing a novel algorithm, namely, Deep Reinforcement Learning (DRL)-based 5G RAN TDD Pattern (DRP), which allows deriving the UL/DL pattern of TDD frames according to the existing cell traffic whatever it is DL or UL dominant. DRP monitors the DL and UL traffic and derives the percentage of the frame (number of slots) dedicated to UL and DL, aiming to avoid the overflow of DL and UL buffers to guarantee the optimal quality of service (QoS) whatever the pattern of traffic, UL or DL dominant. In this demo, we leveraged O-RAN architecture by executing the DRL inference at the near RT RIC as a xApp, while the dynamic TDD mechanism is implemented in OAI. xTDD Service Model (SM) is introduced to plug DRP in OAI following the E2SM standard [5]. xTDD xApp receives UL and DL buffer fullness ratio from the base station via E2 indication messages periodically and executes DRP to derive the TDD pattern. Finally, xTDD sends the TDD pattern via E2 control message to the base station. The latter will update the cell configuration allowing to ensure SON function for D-TDD.

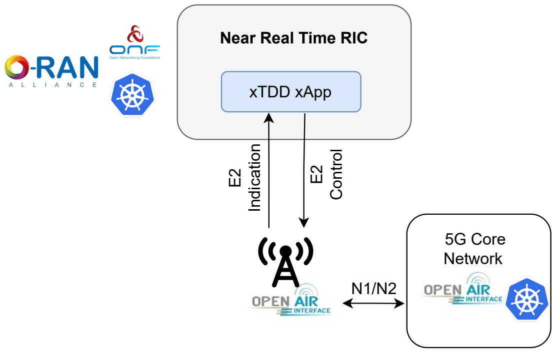

Figure 1: O-RAN aligned architecture of xTDD

2 SYSTEM DESIGN AND IMPLEMENTATION

In this demo, we improved the design introduced in [1] by aligning it with O-RAN architecture as depicted in Figure 1. xTDD xApp executes the DRP agent, responsible for deriving the TDD pattern. xTDD receives E2 indication messages each T ms from the base station containing the DL and UL buffer status. The DL buffer status is extracted from the RLC layer by summing the amount of remaining data (in bytes) after each scheduling process over all the Logical Channels (LC). In contrast, the UL buffer status is estimated at the MAC layer by summing the Buffer Status Report (BSR) MAC Control Element (CE) received over all the LCs. xTDD derives the TDD pattern taking buffer status history as input, and sends a E2 control message containing the ratio of UL slots. The xTDD SM at the base station updates the TDD configuration of the cell at the first slot of the next TDD period upon the reception of the E2 control message.

At the gNB side, we have implemented D-TDD in OAI by: (i) removing the TDD pattern information from the periodic SIB1 broadcast to UEs, which lead the latter to figure out the direction of the slot dynamically via the Downlink Control Information (DCI); (ii) changing the MAC and PHY layers context to change the TDD pattern at gNB. (iii) sending multiple DCI with different K2 parameters in order to schedule multiple UL slots in the same DL slot (to enable more UL slots than DL slots).

At the RIC side, we have implemented DRP using the Deep Deterministic Policy Gradient (DDPG) Algorithm [6]. The DRL hides the complexity of the environment, which helps DRP to make efficient and quick decisions that adapt according to the traffic patterns. We define the DRP design by:

State: The DRP agent considers $\mathcal{K}$ previous observations before taking any action. Each observation depicts the buffer fullness ratio of both DL and UL.

Action: The DRP agent has only one continuous action a_t that presents the percentage of slots that should be reserved for the UL traffic.

Reward: The DRP agent receives a positive reward when the buffers do not exceed their threshold (i.e., the maximum size of the buffers). Moreover, the emptiest the buffers are, the highest reward becomes. The agent receives a penalty when one of the buffers exceeds its capacity. This strategy will enforce the DRP agent to keep all buffers (DL and UL) empty as much as possible and prevent their overflow, which positively impacts the QoS.

We leveraged μonos SD-RAN to create the xTDD SM and xApp and integrate it with μonos RIC. xTDD xApp gains the ability to learn with time and adapts to different and unseen situations. We have designed xTDD to be lightweight to ensure real-time interaction with OAI. Also, we have designed xTDD to ensure generality and then work in an unseen environment. xTDD has been designed in a way to work independently from the number of slots (which makes xTDD suitable for multiple numerologies) and the number of UEs. Further, it considers the variation and correlation in the buffer states to predict the traffic patterns.

3 DEMONSTRATION

3.1 Equipment and Settings

Our setup is composed of (i) Two machines with 36 CPUs, each CPU is an Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz. One machine is used to run gNB based on OAI (based on commit d49db79). OAI is connected to AW2S Radio Unit (RU). The second machine is used as a single node cluster based on Kubernetes. It hosts the 5G Core Network based on OAI, and the SD-RAN services [4] of μonos RIC. (ii) two laptops, each one connected to a Quectel RM500Q-GL module, considered as 5G UEs.

3.2 Experiment Scenario

We connect two UEs to the 5G network. The base station uses numerology 1 and a TDD period of 5ms. For instance, a TDD period has 10 slots. Using Iperf tool [2], we generate traffic for UL and DL following different scenarios with different rates. (i) UL traffic only: we will observe that xTDD selects more UL slots; (ii) DL traffic only: we observe that xTDD selects more DL slots; (iii) UL and DL traffic: we observe that the base station changes the pattern dynamically to avoid buffers overflow and to satisfy the requested data rate from Iperf clients.

ACKNOWLEDGMENT

This work was partially supported by the European Union's Horizon 2020 Research and Innovation Program under 5G!Drones project (Grant No. 857031) and MonB5G project (Grant No. 871780).

REFERENCES

[1] Miloud Bagaa, Karim Boutiba, and Adlen Ksentini. 2021. On using Deep Reinforcement Learning to dynamically derive 5G New Radio TDD pattern. In *GLOBECOM 2021*.
[2] Jon Dugan, Seth Elliott, Bruce A. Mah, Jeff Poskanzer, Kaustubh Prabhu. 2022. *Iperf: The ultimate speed test tool for TCP, UDP and SCTP.* https://iperf.fr/
[3] Florian Kaltenberger, Guy De Souza, Raymond Knopp, and Hongzhi Wang. 2019. The OpenAirInterface 5G new radio implementation: Current status and roadmap. In *WSA 2019, 23rd ITG Workshop on Smart Antennas, Demo Session, 24-26 April 2019, Vienna, Austria.*
[4] Open Networking Foundation (ONF). 2022. *micro onos based RIC.* https://docs.sd-ran.org/sdran-1.4/introduction.html
[5] Open RAN alliance. 2022. *O-RAN specifications.* https://www.o-ran.org/specifications
[6] OpenAI. 2022. *Deep Deterministic Policy Gradient.* https://spinningup.openai.com/en/latest/algorithms/ddpg.html