WiFi Dynoscope: Interpretable Real-Time WLAN Optimization

Jonatan Krolikowski, Ovidiu Iacoboaiea, Zied Ben Houidi, Dario Rossi
Huawei Technologies France SASU
{jonatan.krolikowski, ovidiu.iacoboaiea, zied.ben.houidi, dario.rossi}@huawei.com

***Abstract*—Today's Wireless Local Area Networks (WLANs) rely on a centralized Access Controller (AC) entity for managing a fleet of Access Points (APs). Real-time analytics enable the AC to optimize the radio resource allocation (i.e. channels) online in response to sudden traffic shifts. Deep Reinforcement Learning (DRL) relieves the pressure of finding good optimization heuristics by learning a policy through interactions with the environment. However, it is not granted that DRL will behave well in unseen conditions. Tools such as the WiFi Dynoscope introduced here are necessary to gain this trust. In a nutshell, this demo dissects the dynamics of WLAN networks, both simulated and from real large-scale deployments, by (i) comparatively analyzing the performance of different algorithms on the same deployment at high level and (ii) getting low-level details and insights into algorithmic behaviour.**

I. Introduction

IEEE 802.11 WLANs are the ubiquitous communication medium for a variety of corporate and campus networks. For many medium to large scale deployments of Access Points (APs), the high density makes the allocation of scarce radio resources even more challenging. Luckily, the emergence of big data analytics allows for dynamic WLAN *channel* and *bandwidth* reconfiguration policies that are capable of adapting to sudden demand shifts. This is mediated by a centralized Access Controller (AC) that is responsible of APs real-time monitoring and control.

Such channel management is a combinatorial optimization problem [1] where current state of the art solutions make use of heuristics [2]. Recently, Deep Reinforcement Learning (DRL) techniques [3] have fashioned a new trend in solving such combinatorial optimization problems. However, since training on real networks is impossible –as this would be utterly slow and cause serious operational disruptions– the only option is to train on simulators. Such learning approach faces thus two challenges: (i) as for any algorithm, the DRL solutions must generalize, i.e., exhibit good *transfer learning* capabilities across different networks and scenarios; (ii) additionally, unlike simple man-made heuristics that are interpretable by design, DRL decisions are inherently opaque and lack *explainability*, an important factor that ultimately affects deployability of the solution.

To counter the above two problems, as a side-product of our research on dynamic WLAN Radio Resource Management (RRM), we developed the *WiFi Dynoscope*, an interactive dashboard that allows experts to: (i) compare performance of dynamic DRL reconfiguration policies to traditional heuristics and (ii) gather interpretable insights on the decisions of each algorithm.

Such comparison can be seamlessly done on simulated traffic, as well as from real data collected in real-time from operational large scale WiFi networks, where each day we let the AC employ a different algorithm for performance benchmarking. In particular, we always oppose two dynamic WLAN RRM algorithms at a reconfiguration timescale of 10 minutes, that we pick among two families:

- *ML-based* our DRL solutions that Dynoscope helps interpreting.
- *Heuristics* among which, a static configuration optimized for peak AP traffic and never changed throughout the day; our dynamic local search heuristic that is interpretable out of the box; and finally TurboCA [2], the Cisco Meraki algorithm as the current dynamic RRM state of the art.

The remainder of this extended abstract overviews the capabilities of the platform and outlines the demonstration.

II. WiFi Dynoscope

A. System overview

The WiFi Dynoscope is composed of a *front-end*, which we will show in the demonstration, and a *back-end*. To put it simple, the front-end is an interactive web-interface, built over `Voila` and `Plotly` and fed by aggregated data from our backend. The latter is a pipeline that continuously collects telemetry data leveraging `Kafka`, `Logstash`, and `Spark` , and pushes them into an `Elastic Search` engine. It has also an interface built on `Kibana` technologies to visualize simple statistics about the live network.

This choice allows to seamlessly support both *simulated data* (fed directly), as well as *real operational data* (exported by the products as kafka streams). For the latter, real tests are performed over weeks worth of real traffic using a busy 34 AP WLAN deployed in a Huawei campus canteen in Nanjing, attended daily by thousands of employees. Each day, a different algorithm of those listed above is run in the AC. For repeatable simulation over statistically similar scenarios, we use the same topology and traffic load profiles of the APs. We also generate synthetic challenging workloads.

As such, supporting both simulated and real data gathers the best of both methodologies. On the one hand, simulated data allows to compare different algorithms on the same exact scenario and input (since traffic pattern is controlled), which

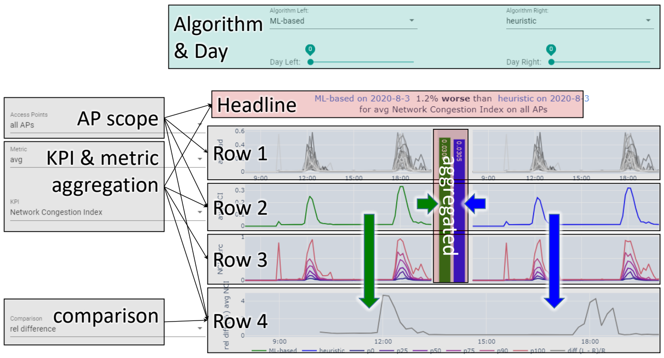

Fig. 1. Performance at a glance: Side-to-side comparison of different algorithms execution on multiple KPIs (the figure annotates the real dashboard screenshot)

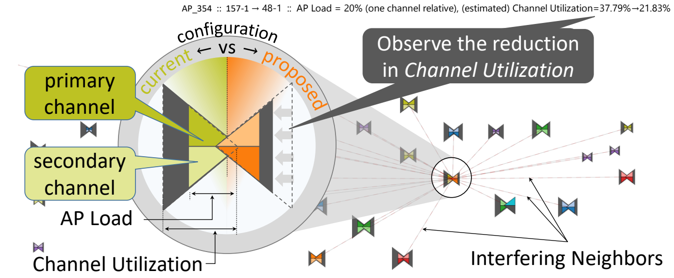

Fig. 2. Operational view: Real-time map of APs configuration load and interference (the figure annotates and zoom in a real dashboard screenshot)

is desirable from a scientific standpoint, but that suffers from lack of realism. On the other hand, the use of real production network allows to statistically compare the execution of different algorithms on the same network, but on different input (since the traffic patterns change from day to day).

We now mainly discuss the front-end, through which the INFOCOM attendees will be able to interact.

B. Performance at a glance

The Performance View (Fig. 1) allows to absolutely evaluate and relatively compare the configuration performance. In case of simulated traffic, it is possible to display side by side the results of two competing algorithms on the same input data. In case of real traffic, the comparison is done over multiple aggregated days.

The user can select two *algorithms* to be compared, and sliders allow to cycle through the different data instances (highlighted in green in the picture). Several Key Performance Indicators (KPIs) are included to fully elucidate the comparison (areas highlighted in gray in the picture). The comparison is carried out using KPIs such as reconfiguration cost, interference, channel utilization, as well as the network congestion index. The latter is our in-house metric which uses an approximation of user throughput to derive an overall network performance score. For instance, load profiles of the APs are always reported in the first row, the 2nd and 3rd row report KPI of interest respectively aggregated over the whole network (2nd row), or in percentiles over all APs (3rd row). Aggregation is flexible in that any metrics can be used (average, median, higher moments or percentiles). Finally, 4th row compares directly the performance (e.g. relative or absolute difference) of the algorithms. The central bar plot completes the visualization (highlighted in red in the picture), aggregating the results in the second row over the entire day. To ease interpretability, a short textual headline summarizes this aggregated output clearly stating and quantifying the winner (also highlighted in red in the picture).

C. Detailed View

The Operational View (see Fig. 2 right) displays a spatial view of the WLAN topology for any desirable reconfiguration moment, that can be selected by the user (by simply hovering over the point in time to zoom on). In this view, hovering over a specific AP highlights the AP neighboorhood properties and brings a zoomed view of the AP configuration. Each AP is represented as a "butterfly" symbol, where the left and right sides stand for the previous and the new configurations, respectively. In particular:

- The *channel configuration* translates to the colors of the inner triangles (here at most two channels are bonded),
- The *AP load* is reflected in the size of the inner triangles.
- The (estimated) *channel utilization upper bound* is proportional to size of outer triangles (dark grey),

As we will demonstrate, this Operational View aids in analysing the behaviour of the a policy, which is especially useful for ML-based algorithms. Gathering intuition, and addressing the explainability of the algorithm behaviour allows network engineers to confidently approve their deployment in production environments.

D. Demo outline

Ultimately, WiFi Dynoscope allows to directly compare RRM algorithms in a scientific manner, as well as gather statistically relevant performance comparison of the real deployment over time. The demo will guide the user through (i) an apple-to-apple comparison of algorithms in simulated settings, (ii) a detailed view of the operations of each algorithm and (iii) a statistically significant comparison of algorithms based on real results from our deployment.

REFERENCES

[1] T. Park and C. Y. Lee, "Application of the graph coloring algorithm to the frequency assignment problem," *Journal of the Operations Research society of Japan*, 1996.

[2] A. Bhartia, B. Chen, F. Wang, D. Pallas, R. Musaloiu-E, T. T.-T. Lai, and H. Ma, "Measurement-based, practical techniques to improve 802.11ac performance," in *Internet Measurement Conference*, IMC, 2017.

[3] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems 28*, 2015.