Orchestration and Reconfiguration Control Architecture

SHOWCASE 3

DISTRIBUTED END-TO-END NETWORK SLICING AND ORCHESTRATION

GOALS

- Demonstrate the deployment of customised and isolated end-to-end network slices.

- Demonstrate how end-to-end network slicing set up can be used to support different types of services with diverging service requirements on top of a shared physical network infrastructure.

CHALLENGES

One of the main challenges of this showcase is the coordination on the resource allocation among different network segments for deploying end-to-end network slices:

- First of all, it was necessary to virtualise network segments for creating customised and isolated network segment slices, e.g., a virtual radio access network (RAN), a virtual transport network (TN), and a virtual core network (CN).

- Then, decomposing the end-to-end network requirements per network segment, allowing the delegation of the resource management to separate specialised orchestrators, tailored for the particularities of each network segment.

- Finally, achieving a cohesive resource allocation across multiple network segment slices to ensure a consistent end-to-end QoS for the network slices.

CONCEPT

This showcase emulates a real network infrastructure that can be encountered in mobile network deployments. More precisely, we consider a scenario whereby the network provider (NP) can use its physical network infrastructure to offer network slices as a service (NSaaS) — in other words, creating network slices on the fly to support different types of communication services and serve service providers (SP).

- First, we instantiate network slices as a service, reacting to requests from SPs, which contain high-level end-to-end service requirements, e.g., throughput, delay, reliability.

- Then, our highlevel orchestrator, the hyperstrator, maps the high-level end-to-end requirements onto high-level local requirements for the separate network segments, enabling the decentralisation of the decision over the resource allocation and function placement to specialised orchestrators in charge of specific network segments.

- Moreover, our hyperstrator coordinates the deployment of network segment slices and ensures a cohesive and optimised performance across networks segments to guarantee a consistent end-to-end QoS for fulfilling the given service request.

Orca

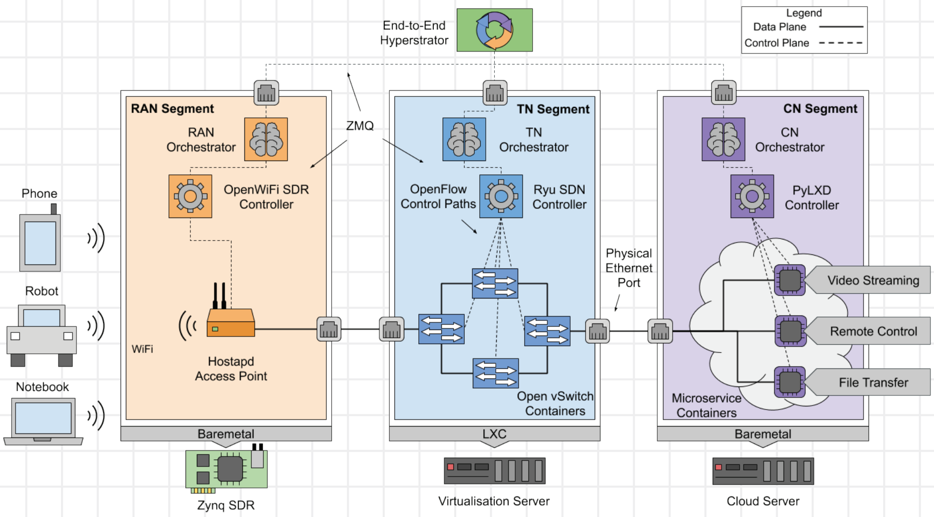
Orchestration and Reconfiguration Control Architecture

SHOWCASE 3

DISTRIBUTED END-TO-END NETWORK SLICING AND ORCHESTRATION

DEMO SET-UP

The demo consists of an experimental end-to-end network infrastructure comprised of three networks segments: a CN, a TN, and a RAN. We use this network infrastructure for supporting three distinct types of services: best-effort remote file storage for notebooks, high-throughput video streaming for handhelds, and low-latency remote vehicle control for autonomous cars; each assigned to a specific end-to-end network slice. The details are shown in the figure below.

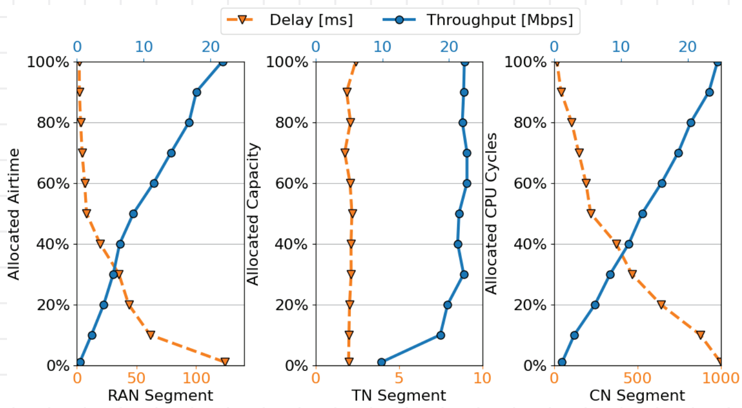


RESULTS

- We show the time required to setup a NS coordinated by the hyperstrator, the setup time is broken down to three segments (RAN ~60 ms, TN ~20 ms and CN ~2000 ms), in total it takes 2.2 second to initiate a NS through the hyperstrator.

- We vary the resource allocation of each network segment, while maintaining maximum resource allocation in other segments. We observe the end-to-end network performance varies with per segment resource allocation, as shown in the figure on the left.

- Hence by coordination through hyperstrator, network resources are allocated appropriately, we guarantee a coherent end-to-end network performance.



IMPACT

- This showcase demonstrates the coordination across multiple network segments for the deployment of customised and isolated end-to-end network slices, which opens a variety of possible extensions and future works, including modelling and analysis of traffic patterns to leverage statistical multiplexing on the allocation of heterogeneous resources to create network slices.

- In order to virtualise the RAN segment and precisely control air time per NS, we developed a WiFi radio hypervisor on an FPGA and embedded ARM platform, which resulted in an open-source project called openwifi. It is a full stack linux mac80211 subsystem compatible open-source Wi-Fi chip design, which is now publicly available for the research community on github (https://github.com/open-sdr/openwifi).

www.orca-project.eu