Decentralized Workload Adaptive Control Plane for Edge Software-Defined Networked Systems

José Moura[1]

[1] Instituto de Telecomunicações (IT), Instituto Universitário de Lisboa (ISCTE-IUL), Lisboa, Portugal
jose.moura@iscte-iul.pt

Keywords: Decentralized, control, workload, adaptive, edge networked system

From the available literature, we have identified the lack of a simple, low system overload, and distributed solution to manage a dynamic cluster of synchronized controllers, following the control workload trend. Thus, the current work investigates a solution that uses a multicast East/Westbound protocol for orchestrating any number of SDN controllers in a distributed way. Considering scenarios where suddenly the amount of new data flows changes, the number of active controllers is dynamically tunned to efficiently process the varying workload associated to the controlled switching devices, which are in parallel continuously and correctly redistributed among all the active controllers. The proposed solution was evaluated as visualized in Figure1.

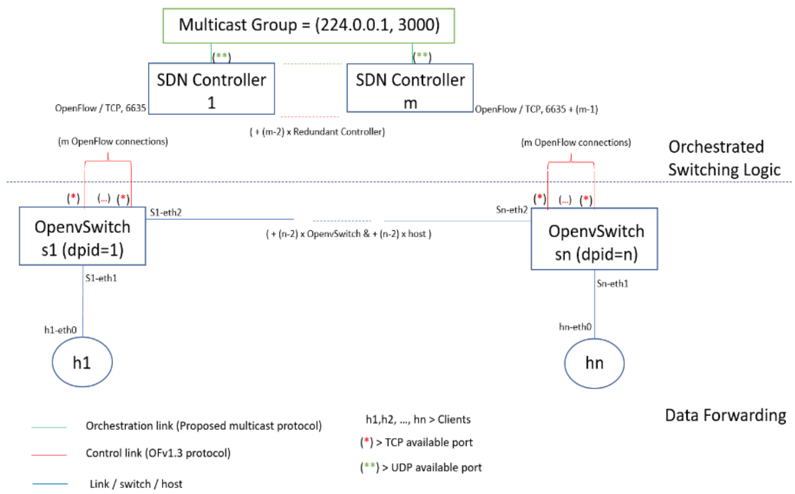

Figure 1. The programmable system under evaluation.

The proposed online function that runs in each controller was tested and the obtained results are visualized in Figure . This function takes a decision of either stopping a controller or launching a new one. Each one of these decisions is made considering the comparison between the online control channel load and a pre-defined system threshold. In the left part of Figure , there is a sequence of actions for stopping a controller in each action, because the *pi_var* is always below the threshold. This sequence for diminishing the number of controllers stops when the system reaches the minimum number of controllers, i.e. two. In the right part of the same figure, it is visible a sequence of actions for launching a new controller in each one of these actions, because the *pi_var* is always over the threshold. This sequence for increasing the number of controllers stops when the system reaches the maximum number of controllers, i.e. ten. We have also observed during the current test that despite the on-demand (de)activation of controllers and their association to switches, the new data flows were not severely degraded. In fact, by configuring a hard timeout of 2s for the flow rules and using a continuous ping between the hosts with the highest number of links among them, we have obtained from the ping statistics the next values: 79 transmissions, 77 received, 9 duplicates and 2.532% packet loss. The sequence of new ping flows with a time interval of 2s was used to increase the number of Packet In messages in the control channel and force the increase on the *pi_var* system variable, which is visible at the right part of Figure 2. In this way, by running the current test, the complete expected behavior of this distributed function was successfully validated.

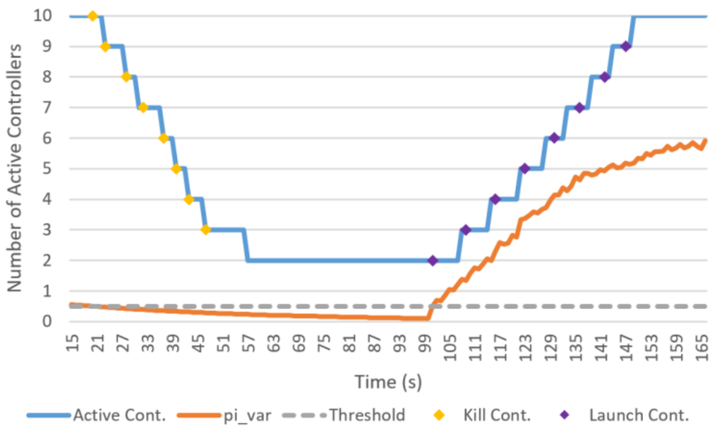

Figure 2. Evaluation of the Distributed Online Function (the *pi_var* and *Threshold* visualized values were multiplied by 10)

As shown by our evaluation, our proposal has evident merits such as, by comparing a unique system variable (*pi_var*) against a pre-defined threshold, the number of active controllers follows the channel control load dynamics. It also guarantees, independently on the number of active controllers, a decentralized orchestration among them. Further study is envisioned: i) system tradeoff between reducing the energy consumption and performance enhancement; ii) distinct control messages from Packet In for accounting workload and balancing it; and iii) add to the exchanged controller ID other useful data namely statistics or topology updates.

Acknowledgment: Jose Moura acknowledges the support given by Fundação para a Ciência e Tecnologia / Ministério da Ciência, Tecnologia e Ensino Superior (FCT/MCTES) through National Funds When Applicable Co-Funded European Union (EU) Funds under Project UIDB/50008/2020 and in part by Instituto de Telecomunicações, Lisbon, Portugal.