

Distributed Network Control for QoS Assurance in Multi-Domain Networks

H. Shakespear-Miles, S. Barzegar, M. Ruiz, and L. Velasco

Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
e-mail: hailey.shakespear@upc.edu

ABSTRACT

The deployment of beyond 5G and 6G networks introduces many new services with stringent Quality of Service (QoS) requirements. Recently machine learning has been shown to be a viable solution in proposing adaptable solutions. However, centralized machine learning based solutions still encounter hurdles in achieving real-time responsiveness due to their need of a global network view. In this paper, we explore a distributed approach aimed at optimizing network performance in real-time scenarios. By using Multi-Agent Systems (MAS), our method targets near-real-time end-to-end delay assurance across diverse network domains, without the need for prior traffic profile knowledge. Evaluated results highlight the effectiveness of our approach in reducing routing costs and ensuring desired end-to-end delay levels.

Keywords: *Distributed Control, Autonomous systems, Deep Reinforcement Learning, Multi-Domain*

1. INTRODUCTION

In order to support the demands of beyond 5G and 6G services, transport networks must evolve to handle increased traffic dynamicity and stricter performance requirements. In fact, such support requires increased levels of flexibility and automation, together with higher priority given to network optimization, security, energy consumption, and cost efficiency. As a result, such infrastructures need to apply Artificial Intelligence (AI) / Machine Learning (ML) techniques [1] to create an automated management to implement data-driven closed control loops. To achieve autonomic networking, Software defined Networking (SDN) control is being augmented with instantaneous data-driven decision-making [2]. This approach is beneficial for many applications that do not require making decisions near real time, like failure management.

In the case of dynamic traffic conditions, centralized decision-making leads to poor resource utilization and high energy consumption. However, precisely because of its centralized location, (near) real-time decision making does not fit well with SDN controllers. In particular, in the case that automation needs to deal with highly dynamic traffic conditions, centralized decision-making leads to poor resource utilization because of long response times. In this work, we focus on flow routing, where decisions need to be made near real-time to optimize resource utilization while ensuring the Quality of Service (QoS) of the flows. It's worth noting that traffic variations may introduce bottlenecks impacting end-to-end (e2e) delay, defined as the time required for transmitting flow traffic between two border packet nodes.

In addition, ML algorithms might be executed as close as possible to the data sources (contrarily to the centralized architecture of SDN) looking at minimizing the amount of data to be conveyed, as well as minimizing the response time. Examples, include the use of Deep Reinforcement Learning (DRL) for the management of the capacity at packet [3] or optical connections (lightpath) [4] in real time, where the capacity of the connections is managed near real-time to adapt to input traffic. Note that packet and optical layers are closely related in multi-layer networks, where links connecting packet nodes are supported by lightpaths. A possible framework for distributed solutions is Multi-Agent Systems (MAS) [5].

This work consolidates findings from two prior studies [6][7] to underscore the intricate nature of routing packet flows across diverse output interfaces for multi-domain networks, emphasizing the dual objectives of maintaining Quality of Service (QoS) standards and optimizing resource utilization. The subsequent sections of this paper delineate our approach and findings. Section II elaborates on the distributed intelligence architecture facilitating near-real-time decision-making in the multi-domain scenario. Section III delineates the construction of the DRL model, including the autonomous flow routing reward function and special considerations that must be made in the multi-domain case. Section IV presents the simulation results, and finally, Section V outlines the conclusions derived from our studies.

2. DISTRIBUTED AUTONOMOUS FLOW ROUTING

Figure 1 sketches the distributed intelligence architecture, where a single node and the centralized SDN controller are represented. Note that in such architecture, we are moving the intelligence from the centralized control plane to the nodes thus resulting in a hybrid centralized SDN control with distributed network intelligence. Agent nodes are able to communicate with each other to exchange data and models for the sake of coordination. Decision-making is performed by every individual agent near real-time (sub-second to few second granularity) based on its own observed data, as well as on the data and models received from other agents. The control plane oversees the overall network coordination, generating necessary guidelines for agents to operate autonomously with the desired degree of freedom. To illustrate this concept, consider the packet layer; packets in a flow follow the route that has been decided from the SDN controller. Route computation is based on the network topology and typically stable

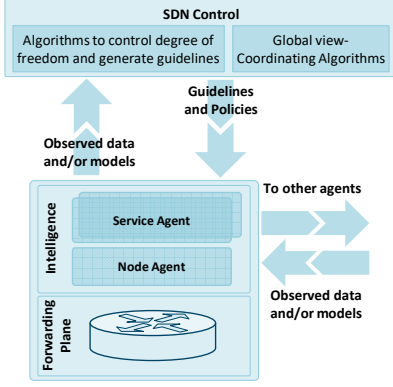


Figure 1: Distributed intelligence architecture

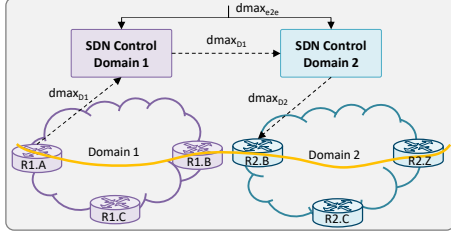


Figure 3: Example of operation under varying d_{max} in multi-domain scenarios.

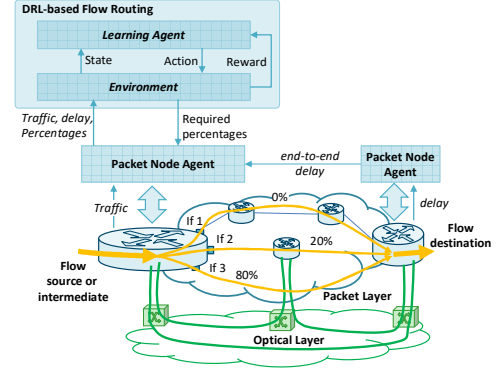


Figure 2: Example of distributed flow routing based on DRL

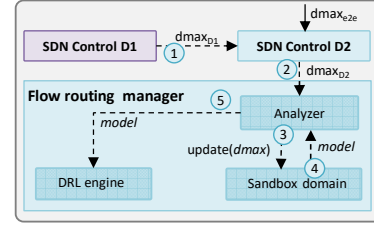


Figure 4: Flow operation in multi-domain scenarios.

unless network conditions alter it. In our approach, the SDN controller gives degrees of freedom to the packet layer by computing a set of routes (including a single one) for every flow that are given to the node agents as guidelines (together with some other parameters), so the actual route is decided by the node agents based on AI/ML models and observed data (e.g., end-to-end delay) and it might be changed near real time. Other intricate tasks, like multilayer issues and failure management, remain centralized, relieving the control plane from immediate operations to prioritize long-term activities. Figure 2 illustrates flow routing within a multilayer scenario, where the packet node agent receives three potential routes from the SDN controller for a given traffic flow. It must use DRL to determine the optimal route or combination to achieve desired QoS while minimizing cost. The figure depicts two interconnected blocks: the former is in charge of learning the best actions to be taken based on the current state and the received reward, whereas the latter is in charge of computing the state based on the observed traffic and current combination of routes (percentages), as well the obtained reward based on the measured end-to-end delay for the selected routes. Multiple sub-flows follow distinct routes, with end-to-end delay measured at the destination and statistics relayed to participating node agents.

A specifically challenging scenario is multi-domain networks, where a packet flow traverses two different administrative domains. Examples, include access networks (fixed or mobile) and metro/core networks. Although the e2e traffic flow consists of two segments, one in each domain, $dmax_{e2e}$ needs to be ensured. Even when each domain works under low to moderate load regime, delay fluctuations are produced as a result of traffic variations, which makes load also variable in time. In this case, the SDN controllers of each domain have received the required $dmax_{e2e}$ for the traffic flow at provisioning time. It is worth noting that if both domains operate without any coordination among them, large capacity overprovisioning is required to absorb delay variations introduced not only by the own domain, but also by the other domains traversed by the flow. In view of that, we assume some sort of coordination among domains. An example is represented in Figure 3. The SDN controller of domain 1 dynamically gets the delay that can be ensured for the segment of the flow ($dmax_{D1}$) and share that value with the SDN controller of domain 2. In response, the SDN controller tunes the requirement of delay for the local segment ($dmax_{D2}$) so $dmax_{e2e}$ is ensured. We expect that overprovisioning can be greatly reduced and e2e delay guaranteed by adjusting domain delay budgets dynamically.

3. DRL OPERATION IN MULTIDOMAIN SCENARIOS

In this section, we detail the reward function for Twin Delayed Deep Deterministic Policy Gradients (TD3) [7]. The state is defined as the ratio traffic over the capacity of the interfaces. In addition, each action is defined as being related to one flow and output interfaces and represents the percentage of flow to be sent through those interfaces. As an example, for one single flow that can be routed through 3 different interfaces, action [50, 20, 30] entails 50% of traffic flow through the first interface, 20% through the second one, and 30% through the third one. In our approach, the computation of states and actions is performed periodically (e.g., every second).

A generic reward function has been defined in Equation 1 with the objective of penalizing the actions causing that some target delay is exceeded and/or increasing the cost of network. In consequence, two reward components

have been considered, to account for the obtained delay (r_{delay}) and for the cost (r_{cost}), where the final reward is defined as follows; parameters α_{delay} and α_{cost} represent the proportion of each of them. Assuming a given maximum delay to be ensured for the flow (denoted D_{max}), the reward related to the obtained delay can be defined in Equation 2, where β is a fixed penalty for violating the maximum delay. Finally, the reward related to the cost of using the output interfaces is related to the percentage of traffic sent through each of them, as well as to the ratio cost capacity of the interface shown in Equation 3. We assume that the capacity of the interfaces, as well as the cost of each interface (which is related to the route to the destination for the flow), D_{max} and β for each flow have been received from the SDN controller.

$$r(t) = \alpha_{delay} \cdot r_{delay}(t) + \alpha_{cost} \cdot r_{cost}(t) \quad (1)$$

$$r_{delay} = \{-\beta - d/D_{max} \text{ if } d > D_{max} \ 0 \text{ otherwise}\} \quad (2)$$

$$r_{cost} = -traffic \cdot \sum_i percentage_i \cdot \frac{cost(i)}{capacity(i)} \quad (3)$$

In order to properly adapt the autonomous operation to multi-domain scenarios, coordination between domains is implemented to satisfy the e2e delay requirement. Figure 4 shows the workflow that implements such coordination. For the sake of simplicity, we assume the scenario in Figure 3, where the flow under study crosses Domain 1 (D1) before entering reference domain D2. Recall that $d_{max_{e2e}}$ need to be ensured and thus, autonomous operation in D2 needs to guarantee such requirement, considering the delay in the previous domain D1.

Then, once operation starts, the controller of D1 asynchronously notifies its maximum delay $d_{max_{D1}}$ to the network controller of Domain 2 (labeled 1 in Figure 4), which computes the requirement for its domain segment as $d_{max_{D2}} = d_{max_{e2e}} - d_{max_{D1}}$ (2). This value is pushed to the flow routing manager, that will work to guarantee such updated $d_{max_{D2}}$ requirement. In particular, the analyzer asks to the sandbox domain for updating to the new d_{max} (3). The sandbox evaluates whether the current model can properly work with the new QoS requirement; otherwise, return a new model (4) to be loaded into the DRL engine (5). Since the sandbox stores monitored $d(t)$ for a given history, it evaluates whether past $d(t)$ measurements are below new d_{max} . If so, no model update is necessary; otherwise, a new model specifically trained for such new d_{max} is loaded. Note that, in both cases, continuous online learning will improve the model by learning from its routing decisions.

4. RESULTS

A Python-based simulator was implemented and realistic traffic flow behavior was accurately emulated following a similar configuration as in [3]. As in Figure 2, we assume that $if1$, $if2$, and $if3$ follow different routes in the multilayer network, so that three different delay behaviors are emulated. The cost of each interface was configured inversely proportional to the expected end-to-end delay through that interface. $\beta=6$, $D_{max} = 0.5$ ms, and compared three different optimization targets by defining different configurations of the tuple $(\alpha_{delay}, \alpha_{cost})$, namely: (0,1) (cost minimization), (1,0) (delay assurance), and (1,1) (multi-objective). Figure 5 illustrates the overall performance under all three cases in terms of maximum delay and cost. It is worth noting that the DRL learned a model producing stable and good-rewarded actions in all the scenarios after 5000 episodes. As seen in Figure 5, the first configuration (0,1) achieves the expected best solution in terms of cost, at the expense of having a large maximum delay. Figure 6 details traffic routing and delay one day post DRL convergence. In Figure 6a, traffic solely utilizes the cheapest interface, resulting in delays consistently exceeding D_{max} , compromising QoS. Conversely, configuration (1,0) in Figure 6b reduces maximum delay below D_{max} , albeit at a notable increase in network cost due to favoring the delay-efficient, expensive interface $if1$. However, penalization of delay violation maintains delay well below the limit. Notably, configuration (1,1) in Figure 6 shows the best performance, with maximum delay below D_{max} and a 58% reduction in network cost compared to (1,0). Detailed analysis in Figure 6c reveals DRL's ability to balance routing between $if2$ and $if3$, yielding delays consistently below D_{max} while minimizing costs by avoiding expensive $if1$. This demonstrates DRL's capacity to converge to diverse solutions for heterogeneous optimization criteria. Additionally, a second study explores various D_{max} scenarios ranging from 0.3 to 2 ms under fixed configuration (1,1).

For evaluating multi-domain performance another simulation scenario was run following [7], where background traffic is not constant in time. at time t_0 and a pre-trained initial model assuming constant background traffic and a given $d_{max_{D2}}$ was used for operation. This model is continuously improved through online learning; note that it now needs to learn the actual characteristics of the input traffic and those of the time-varying background traffic. After some time in operation, at time t_1 the model reaches a stable performance that cannot be significantly further improved. Then, at time t_2 , the D2 SDN controller receives an asynchronous notification from D1 SDN controller updating $d_{max_{D1}}$, which in turn triggers updating $d_{max_{D2}}$ and consequently, the proposed analysis and model update procedure is carried out. Then, operation continues with the new $d_{max_{D2}}$. Online learning might improve the model in operation, which will reach performance stability at time t_3 . Figure 7 summarizes the main results of the simulations in terms of routing cost and delay measured at every of the abovementioned time instants. Specifically, Figure 7a and Figure 7b show two cases, where $d_{max_{D2}}$ is relaxed (from 0.5 to 0.75ms and from 0.25

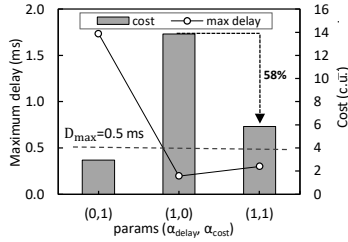


Figure 5: Overall DRL performance

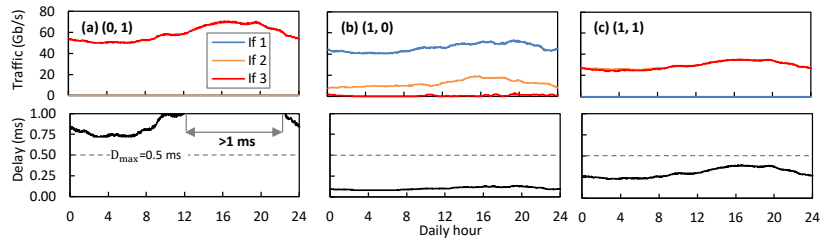


Figure 6: Detailed performance for configuration (0,1)(a), (1,0)(b), and (1,1)(c)

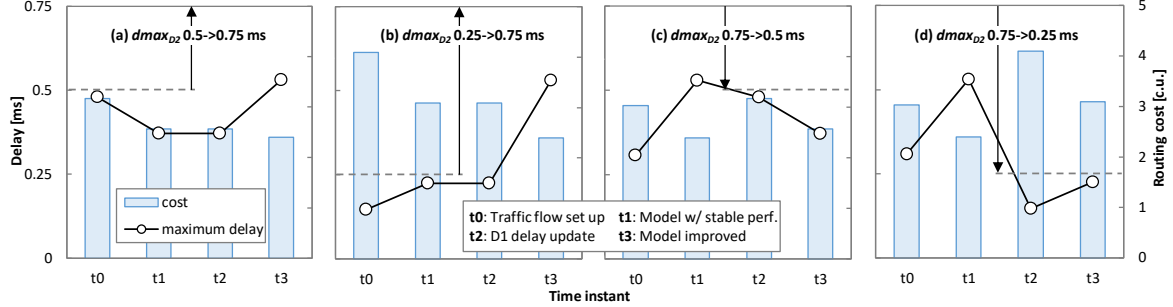


Figure 7: Performance evaluation in multi-domain scenarios with time-varying background traffic

to 0.75ms, respectively), whereas in Figure 7c and Figure 7d $dmax_{D2}$ becomes more stringent (from 0.75 to 0.5 and from 0.75 to 0.25, respectively).

We observe that online learning improves initial pre-trained models even in the presence of time-varying background traffic, since routing cost is reduced in all the cases from t_0 to t_1 , while $dmax_{D2}$ is guaranteed in the whole period $[t_0, t_1]$. In case of Figure 7a and Figure 7b, there was no change in the model in time t_2 because of $dmax_{D2}$ relaxation and hence, performance in t_2 equals that of t_1 . However, a new model was loaded when $dmax_{D2}$ was reduced in time t_2 , and which reduced maximum delay to guarantee the desired QoS performance from t_2 on, as observed in Figure 7c and Figure 7d. Finally, note that regardless the case, the model was improved after $dmax_{D2}$ update, by increasing maximum delay and/or reducing routing cost.

5. CONCLUSION

This paper summarizes a distributed approach leveraging Multi-Agent Systems (MAS) tailored specifically for optimizing network performance in real-time scenarios, with a particular emphasis on ensuring near-real-time end-to-end delay assurance across multiple network domains. The results demonstrate the effectiveness of this approach in reducing routing costs and maintaining desired end-to-end delay levels within multi-domain scenarios. Notably, the adaptability of Deep Reinforcement Learning (DRL) models in achieving heterogeneous optimization criteria. The incorporation of online learning mechanisms enhances model performance, even in the face of time-varying background traffic. Overall, these results underscore the potential of distributed intelligence approaches in addressing the evolving challenges of next-generation networks with stringent QoS requirements in multi-domain environments.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union's Horizon Europe research and innovation programme SEASON (G.A. 101096120), the MICINN IBON (PID2020-114135RB-I00) and from the ICREA Institution.

REFERENCES

- [1] D. Raffique and L. Velasco, "Machine Learning for Optical Network Automation: Overview, Architecture and Applications," (Invited Tutorial) IEEE/OSA Journal of Optical Communications and Networking (JOCN), 2018.
- [2] L. Velasco *et al.*, "Monitoring and Data Analytics for Optical Networking: Benefits, Architectures, and Use Cases," IEEE Network Magazine, 2019.
- [3] S. Barzegar, M. Ruiz, and L. Velasco, "Packet Flow Capacity Autonomous Operation based on Reinforcement Learning," MDPI Sensors, vol. 21, pp. 8306, 2021.
- [4] L. Velasco *et al.*, "Autonomous and Energy Efficient Lightpath Operation based on Digital Subcarrier Multiplexing," IEEE Journal on Selected Areas in Communications, vol. 39, pp. 2864-2877, 2021.
- [5] M. Wooldridge, *An introduction to multiagent systems*, John Wiley & Sons, 2009.
- [6] S. Barzegar, M. Ruiz and L. Velasco, "Distributed and Autonomous Flow Routing Based on Deep Reinforcement Learning," OECC/PSC, Japan, 2022
- [7] S. Barzegar, M. Ruiz and L. Velasco, "Autonomous Flow Routing for Near Real-Time Quality of Service Assurance," IEEE Transactions on Network and Service Management (TNSM) 2023