Distributed Multi-Agent System fed with Telemetry Data for Near-Real-Time Service Operation

Pol González[1], Faris Alhamed[2], Sima Barzegar[1], Francesco Paolucci[3], Juan Jose Vegas Olmos[4], Marc Ruiz[1], and Luis Velasco[1*]

[1] Universitat Politècnica de Catalunya (UPC), Spain; [2] Scuola Superiore Sant'Anna (SSSA), Pisa, Italy; [3] CNIT, Pisa, Italy; [4] NVIDIA Roskilde, Denmark; e-mail: luis.velasco@upc.edu

Abstract: Near-real-time routing decisions on multiple flows will be demonstrated. Decision making is based on precise end-to-end delay telemetry processed by a P4 collector. Distribution of roles will exhibit reduced response times to provide multi-objective operation.

1. Overview

Near-real-time autonomous operation, i.e., when actions are performed in the seconds scale, is a special type of network automation that requires the availability of new technologies and architectures to be developed in the control and data planes. One of the required technologies is that of pervasive in-band network telemetry (INT) to get accurate measurements of relevant metrics, including Quality of Service (QoS), e.g., delay and/or jitter, of the services supported by the network [1]. Recent progress on hierarchical telemetry architectures with distributed intelligence showed the great potential of using such telemetry data for efficient network diagnosis and decision making [2].

Dynamic flow routing is a use case of near-real-time network operation that requires the interaction between telemetry agents and flow agents in charge of making routing decisions [3], which eventually leads to the deployment of a distributed multi-agent system (MAS). Among different routing policies, *multi-path* routing introduces flexibility in the design and operation of the network by allowing operators to split traffic demands into multiple streams that are routed independently of each other to the destination [4]. A possible routing policy is to evenly split each traffic flow among all available routes. However, such a strategy might fail under dynamic network conditions that can generate congestion in some routes and consequently, lead to flows QoS degradation [3]. Moreover, routes might have different utilization costs, and hence, the percentage of traffic sent through each route is a complex decision that needs to be dynamically tuned in order to meet robust QoS performance with overall minimum cost.

In this demonstration, we will showcase the near-real-time operation of an optical packet network controlled by a distributed MAS. The MAS combines: *i*) pervasive INT agents supported on P4-based components [5]; and *ii*) multi-flow routing agents that are used to dynamically adjust multi-path flow routing policies in the packet nodes with the objective to guarantee the target QoS performance. Hence, flow routing operation is controlled by a set of heterogeneous agents that are fed with telemetry data collected from P4 switches. The systems in the demonstration are being developed within the Horizon Europe DESIRE6G project [6] and will be deployed in a distributed federated testbed including the CNIT/SSSA (ARNO testbed) in Pisa (Italy) and the UPC testbed in Barcelona (Spain).

2. Innovation

This demonstration will showcase innovative capabilities and functionalities of P4-based switched networks [5]. In particular, the application of P4 INT telemetry collectors to measure and aggregate per-flow and per-route QoS is a key innovation to enable multi-path routing. Moreover, this demonstration will showcase the operation of a distributed MAS system combining different types of agents (telemetry and flow routing). Note that this architecture focuses specifically on near-real-time control as it greatly reduces response times, while liberating the centralized software-defined networking (SDN) controller from such tasks. Finally, coordination among flow agents making near-real-time flow routing decisions of different flows is required, which highlights the benefits of the distributed MAS.

3. OFC Relevance

This demonstration aims at attracting the attention of the OFC audience interested in novel solutions for network intelligence and autonomous network operation, which are the two main topics that will be covered by this demonstration. Note that such solutions have the potential to meet the requirements from novel beyond 5G and 6G services, which will require bounded delay assurance even under highly dynamic network conditions. Thus, being able to make near-real-time decisions can significantly reduce capital and operational costs for network operators.

Specifically, the demonstration aims at answering questions such as whether a distributed MAS fed with INT can efficiently make near-real-time network operational decisions. Still nowadays, there are network operators, vendors, and researchers who are skeptical about the practical use of autonomic networking. With this demonstration, we aim

The research leading to these results has received funding from the Smart Networks and Services Joint Undertaking under the European Union's Horizon Europe research and innovation programme under G.A. No. 101096466 (DESIRE6G) from the MICINN IBON (PID2020-114135RB-I00) projects and from the ICREA Institution.

at providing a case of success, as well as to foster discussions around aspects such as the feasibility of their deployment in brownfield scenarios, scalability, and deployment cost, just to mention a few.

4. Demo Content & Implementation

This demonstration aims at experimentally assessing the feasibility of a distributed MAS fed with telemetry measurements to perform near-real-time flow routing operation. For illustrative purposes, Fig. 1 shows an example where several traffic flows (*Fi* to *Fk*), each following a multi-path routing strategy, enter and leave a network at different packet nodes. In the example, traffic flow *Fi* (from R1 to R5) can follow three different routes, where *p1* and *p2* are multi-hop paths on the packet network, whilst *p3* uses an optical bypass connecting R1 and R5 through the underlying optical network. As in [3], we assume that traffic flows are *splitable*, i.e., they consist of a large number of sub-flows that can be routed independently. The objective is to find the flow routing policy that balances the incoming traffic of the flows among the available paths, so to ensure per-flow QoS (specifically e2e delay). Such routing policy varies as a function of the incoming traffic of the flow and the network conditions, i.e., the traffic of the rest of the traffic flows in the network. Therefore, the routing policy decision making process is continuously carried out based on the incoming traffic and the e2e delay measurements that allow evaluating the quality of the decision making. Note that the state of the network is known, and it is indirectly represented by e2e delay measurements for the traffic flow. In this demonstration, we rely on the INT functionality provided by the P4 switches to measure packet delay. Specifically, a P4 collector that collects, aggregates, and provides statistics of the delay measured by the switches supporting the traffic flows will be showcased [7]. Once the P4 collector preprocesses the QoS measurements, they are sent to a telemetry agent, which is in charge of producing flow telemetry statistics that are sent to the flow agent deployed at the source node, where flow routing policy decisions are made.

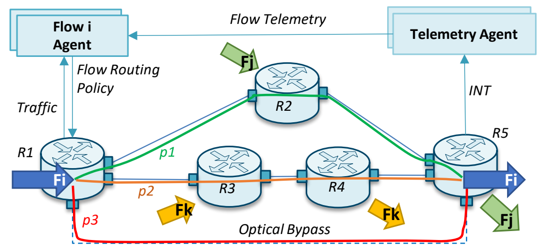

Fig. 1: Illustrative Example

Fig. 2 presents the federated testbed and the experimental setup to be deployed for this demonstration, which reproduces a 5-node network scenario similar to that in Fig. 1. The packet-optical network will be deployed at the CNIT/SSSA ARNO testbed, whereas the distributed MAS is deployed at UPC premises. Five native in-kernel P4-programmable software switches (P4-NIKSS) will be deployed and connected to create the packet network. P4 switches are interconnected among them using 10G interfaces. P4 switches run on a separate server to maximize performance. Specifically, Intel(R) Xeon(R) Gold 6238R CPU @2.20GHz server with 256GB RAM are used. An additional P4-based software process is used as telemetry data collector, which receives INT data from the P4 switches [7]. In order to reproduce network congestion and selectively introduce delay in some of the paths of the traffic flows, a Juniper M10i router equipped with 1G interfaces is connected to switch R1, as depicted in the inner graph in Fig. 2. Selected flows are forced to traverse the Juniper router before route switching, thus generating different congestion scenarios. Regarding the optical layer, the optical bypass between R1 and R5 is based on a pre-programmed packet-optical whitebox employing 10G and 100G optical pluggables. The pluggables are connected to an Optical Line System (OLS) by means of an arrayed waveguide gratings multiplexer. The OLS is composed of three 80km fiber spans and line optical amplifiers. The optical connection between the two pluggables is pre-established and enforced locally by OpenConfig-enabled SDN agents co-located with the SONiC node operating system. Finally, a Spirent SPT N4U equipped with 10G ports is used as packet generator for two different traffic flows (*f1* and *f2*), both from R1 to R5. In our case, flows are identified by source and destination IP. For validation purposes, the Spirent tool is also used as flow traffic sink. In the demonstration, R1 will measure per-flow traffic entering the switch, before being forwarded through the different paths. In particular, three alternative paths are considered: *p1* (R1-R2-R5), *p2* (R1-R3-R4-R5), and *p3* (R1-R5, using the optical bypass).

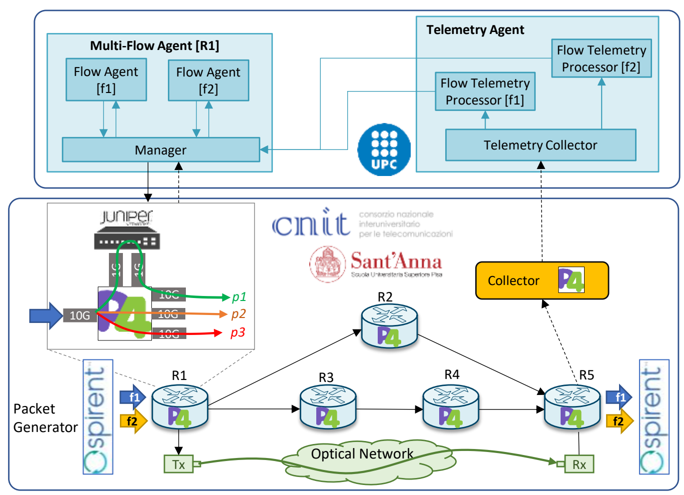

Fig. 2: Federated testbed setup for the demonstration

The distributed system consists of several interconnected software components. Specifically, the *telemetry agent*, which is part of the distributed telemetry architecture [2], includes: *i*) a telemetry collector that periodically receives telemetry data from the P4 collector; and *ii*) a per-flow telemetry processor. The role of these telemetry components is to compute the required measurements and statistics that characterize the current QoS of the traffic flow, from the received INT telemetry. In addition, *flow agents* are grouped in a single module per location named *multi-flow agent*. In this case, the multi-flow agent at R1 includes flow agents for both *f1* and *f2*. Note that one single flow agent makes routing decisions for each traffic flow. The *manager* running inside multi-flow agents has the role of collecting and distributing flow telemetry data and local input traffic data to the flow agents, as well as to push the flow routing policies computed by flow agents to the P4 switch. The demonstration will rely on REST API interfaces between agents and P4 systems. MAS agents and components (including interfaces) are implemented in Python 3.10.4 and run inside Docker containers running on 2 separated VMs with Ubuntu Server 22.04 LTS as operating system.

The workflow to be demonstrated for traffic flows *f1* and *f2* is outlined in Fig. 3. In addition, Fig. 4 presents examples of some selected messages exchanged (identified with the labels in the workflow). Let us assume that flow routing policies are configured at switch R1 (0 in Fig. 3) to enable multi-path routing. Input flow traffic measurements are collected and sent periodically to the local multi-flow agent for every traffic flow entering in the network (1); the messages contain both packet and bit count for every traffic flow (see the details in Fig. 4). Packet delay measured along the path is reported though INT messages to the P4 collector (2), which performs aggregations and computes statistics, e.g., min, average, and max of delay and jitter. Periodically, the P4 collector sends the computed delay statistics to the collector in the local telemetry agent (3). In addition, delay statistics are reported to a centralized telemetry system that stores them in a time series database (not shown in the workflow in Fig. 3). The aggregated QoS telemetry message includes per-path statistics for every flow (Fig. 4). The received statistics are then processed by each of the flow processors (4) that computes per-flow QoS telemetry measurements and send them to the corresponding flow agent (5). When flow QoS statistics are received in the multi-flow agent, the manager triggers its analysis by pushing both traffic and QoS measurements to the corresponding flow agent (6). In our previous work in [3], a pre-trained reinforcement learning model made routing decisions, with the operational objective of guaranteeing that the e2e delay does not exceed a configured threshold (QoS requirement), while minimizing the use of the optical bypass. However, to be able to interact with the system, in this demonstration, we will use a parameterized deterministic algorithm (7), so different routing policies can be manually forced by changing the values of some parameters, e.g., to force using the optical bypass. Routing decisions are gathered by the manager that eventually sends them to the P4 switch (8). The details of message 8 in Fig. 4 show that a routing policy is defined as the percentages of input traffic to be routed through each of the routes. Internally, the P4 switch translates this policy into flow rules to efficiently perform packet forwarding according to defined percentages.

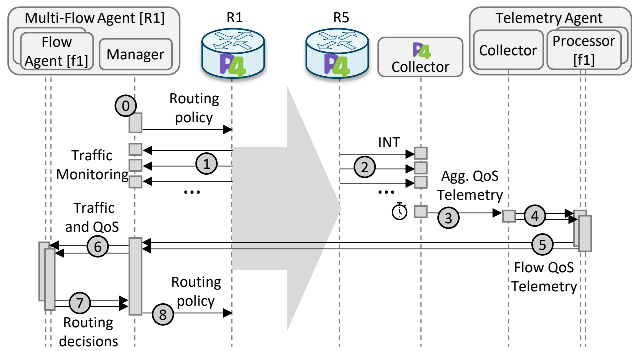

Fig. 3: Near-real-time operation workflow to be demonstrated

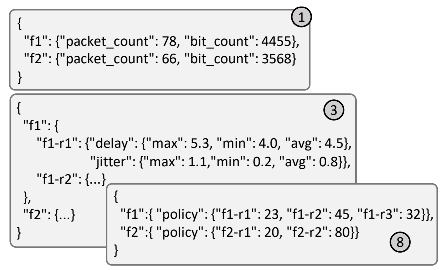

Fig. 4: Selected exchanged messages

The illustrated workflow will be used to demonstrate different traffic scenarios, including: *i*) sharp traffic increase in one of the flows, leading to temporary QoS degradation (peak of delay) in both flows; and *ii*) gradual traffic increase in both flows. In the demonstration, autonomous and fixed routing policies will be demonstrated to show how algorithms can anticipate decision making and avoid congestion, leading to robust high QoS achievement. A Web interface will facilitate the interaction of the attendees with the system, so they can modify the type and configuration of routing policies. Moreover, a Grafana dashboard will be provided to visualize the resulting QoS performance of the flows and observe the impact of routing on flows QoS. In addition, traffic collected by the Spirent sink will be visualized for demo validation purposes.

References

[1] L. Velasco *et al*., "Pervasive Monitoring and Distributed Intelligence for 6G near Real-Time Operation," EuCNC 2023.
[2] L. Velasco *et al*., "Distributed Intelligence for Pervasive Optical Network Telemetry," IEEE/OPTICA JOCN, 2023.
[3] S. Barzegar *et al*., "Distributed and Autonomous Flow Routing Based on Deep Reinforcement Learning," in Proc. PSC, 2022.
[4] J. Rischke *et al*., "QR-SDN: Towards RL States, Actions, and Rewards for Direct Flow Routing in SDN," IEEE Access, 2020.
[5] F. Cugini *et al.*, "Telemetry and AI-based Security P4 Applications for Optical Networks," IEEE/OPTICA JOCN, 2023.
[6] Deep Programmability and Secure Distributed Intelligence for Real-Time E2E 6G Networks project. [On-line] https://desire6g.eu/
[7] F. Alhamed *et al.*, "P4 Telemetry Collector," Elsevier Computer Networks, 2023.