

Demonstration of Integrated Control Plane for Service Flow Provisioning in Time Sensitive Networks

M. Ruiz^{1*}, A. Pagès¹, F. J. Moreno², A. Buendía², F. Agraz¹, S. Spadaro¹, and L. Velasco¹

¹ *Universitat Politècnica de Catalunya, Spain;* ² *E-lighthouse Network Solutions, Spain*

**e-mail: marc.ruiz-ramirez@upc.edu*

ABSTRACT

One of the most challenging tasks that autonomous network control systems need to face is to provision end-to-end packet flows spanning several domains. With the advent of time sensitive networking, the co-existence of services and infrastructure with and without time sensitive capabilities adds more complexity to the end-to-end flow management. In particular, avoiding improper provisioning decisions is key to guarantee stringent quality of service requirements. In this demo, we will showcase an integrated control plane for provisioning flows spanning through domains with and without TSN capabilities. The decision making process will be supported by a digital twin in charge of estimating expected performance indicators before path provisioning acceptance.

Keywords: Time Sensitive Networks, SDN-based Network Control, Digital Twinning

1. OVERVIEW

The advent of beyond 5G services and use cases is pushing telecom operators to adopt novel control and orchestration solutions for guaranteeing the increasing need of stringent end-to-end (e2e) requirements in terms of Quality of Service (QoS) assurance. In particular, packet flows typically from heterogeneous services span through different technological and administrative domains, which requires the adoption of novel network infrastructure management architectures enabling e2e autonomous network operation [1]. Among different traffic classes, time-sensitive (TS) packet flows are those demanding most stringent key performance indicator (KPI) requirements in terms of maximum delay and maximum jitter. Recent advances in network components with time-sensitive networking (TSN) capabilities are paving the way to support those TS services [2]. However, both TS traffic flows and TSN capable infrastructure must co-exist and co-managed with non-TS services and legacy network infrastructure without TSN capabilities. This fact imposes challenging requirements to the control of that heterogeneous network infrastructure.

Software Defined Networking (SDN) paradigm offers the potential and flexibility to manage the provisioning of TS flows in TSN capable infrastructures. Indeed, SDN represents the natural option to enable easy integration of TSN infrastructures with SDN-controlled non-TSN legacy networks. However, the strict conditions of TS services require extending capabilities to achieve e2e KPIs fulfillment success. In this regard, the use of a Digital Twin (DT) can be of paramount importance to estimate the performance of traffic flows near-real-time [3]. DTs can model service traffic and queues behavior in packet nodes, thus building emulated networks where KPIs can be estimated and used for intelligent provisioning decision making.

This demo will showcase an integrated e2e control plane solution for the provisioning of TS and non-TS flows in heterogeneous networks composed by domains with and without TSN capabilities. The demo will require the deployment of several novel control plane elements that interact during the flow provisioning phase. Upon the reception of a new flow request, a provisioning workflow is triggered, which finds a new path for the request spanning multiple SDN-controlled domains, as well as a detailed KPIs analysis provided by a DT. This KPIs estimation, which is provided for both new request under evaluation and already established flows, can be easily and explicitly processed to decide whether the new request can be established while guaranteeing committed QoS.

2. INNOVATION

The demo presents an integrated control plane architecture for e2e TSN-based service provisioning, which is a major innovation. Specifically, the same controller is used for the control and configuration of multiple TSN-enabled data plane technologies, including Ethernet and WiFi, and materializing service provisioning over multiple technological domains. This not only provides a more compact architecture, reducing the overall messaging burden, but also brings the benefits of providing an aligned configuration across technological independent TSN domains, eliminating configuration incompatibilities and optimizing the TSN flow installation. Moreover, the active KPIs assessment during the provisioning phase allows intelligent decision making that avoids accepting new requests that affect the committed performance of already established flows.

3. DEMO CONTENT & IMPLEMENTATION

A. Architecture overview

Figure 1a presents the considered network scenario that covers different domains including network nodes with and without TSN capabilities. This network supports packet flows of different service classes, which can be classified into two main categories: *i) TS packet flows*, which are exclusively supported through TSN-capable devices; and *ii) non-TS packet flows*, which can be supported through both with and without TSN capabilities devices. The example in Figure 1b shows one TSN-capable WiFi access points (AP), three TSN Ethernet switches, and two non-TSN capable packet routers. The network connects one robotic arm, two servers, and a number of users. One TS flow (TS_1) is routed through a path connecting the robotic arm to their controller running in Server A. Additionally, a video flow requiring some QoS performance is established (QoS_1). In another part of the network, the users are connected to Server B, consuming traffic with low priority, e.g., best effort (BE_1).

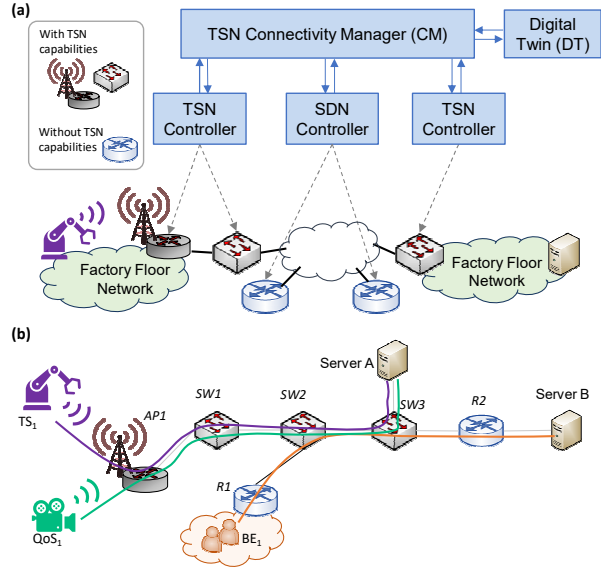


Figure 1: Architecture (a) and example (b)

The architecture in Figure 1a contains a number of key components to deal with the provisioning of TS and non-TS packet flows. For the control of heterogeneous network domains, we rely on SDN controllers with (denoted TSN controller) and without TSN capabilities, which use multiple north-bound (NBI) and south-bound (SBI) interfaces for different purposes, e.g., to program the network devices under their control. The TSN Connectivity Manager (CM) provides e2e control, as well as path computation capabilities to assist SDN-based network controllers. Finally, the Digital Twin (DT) evaluates a set of KPIs of non-TS flows before new (TS or non-TS) flows are deployed. The DT considers non-TS flows with different priorities, e.g., QoS committed and BE. Note that although all flows are served on a particular path defined for each flow, TS flows have specific resources that are reserved along their path, whereas non-TS ones use the remaining resources assigned by their priority.

B. Proposed workflow

Figure 2 illustrates the main workflow to be showcased in this demo. As can be observed, the CM acts as the central orchestrator, interacting with both the TSN and SDN controllers to gather data, and with the DT for KPI assessment to make an informed decision about the service path. The event that triggers the workflow is the arrival of a new e2e service request to process (labeled as 0 in Figure 2). Upon the reception of this request, the CM starts the topology discovery phase, sending a request for topology to TSN controller (1), as well as to SDN controller (2). With the information provided by both controllers, the CM composes the e2e network and computes the multi-domain path for the service (3). In case of finding a feasible path in terms of available resources, the process of evaluating KPIs is triggered. Thus, to ensure that the computed path meets all performance requirements, the CM requests an assessment of KPIs to the DT (4). The DT performs its calculations (5), and returns the estimated KPIs (6), which allows the CM deciding on the e2e service path. Thus, in case of request acceptance, it will issue commands to network controllers in order to configure the data plane accordingly for provisioning the new request. On the contrary, if some KPI is violated, the path is rejected, which could trigger the search and KPIs evaluation of an alternative path excluding conflicting links with the rejected one.

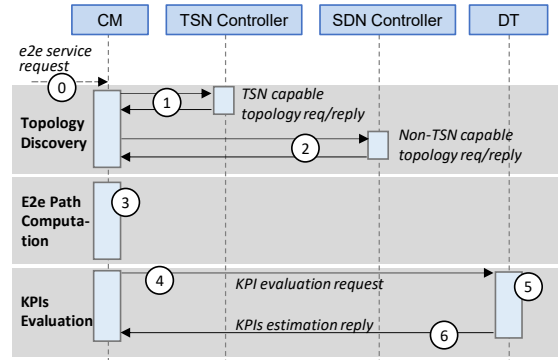


Figure 2: Proposed Workflow

As already mentioned, the CM is a component designed to enhance efficiency in multi-domain e2e orchestration operations. Functioning as the primary interface for verticals to define service requirements, the CM integrates the orchestration and management capabilities in its core module and the path computation and user interaction functionalities given by the E-Lighthouse Network Planner (ENP) [1]. In order to configure the TSN-able data plane, the CM engages with the TSN controller, which is the entity in charge of configuring the physical

C. Implementation and execution details

As already mentioned, the CM is a component designed to enhance efficiency in multi-domain e2e orchestration operations. Functioning as the primary interface for verticals to define service requirements, the CM integrates the orchestration and management capabilities in its core module and the path computation and user interaction functionalities given by the E-Lighthouse Network Planner (ENP) [1]. In order to configure the TSN-able data plane, the CM engages with the TSN controller, which is the entity in charge of configuring the physical

hardware upon service provisioning requests. It acts both as a controller entity for the lifecycle management of data flows at the data plane level as well as an abstraction layer for the CM. Thanks to the capabilities of the multiple NBIs and SBIs, it becomes possible to abstract the technological details of the data plane from the CM, which will only request for the configuration of packet flows according to a topological path plus some TSN service requirements. Then, via informed models the TSN controller will translate said flow into fine-grain configurations at the data plane via the multiple SBIs.

During the provisioning workflow, the TSN controller is contacted by the CM in order to gather information about the topology of TSN domains, as well as the capabilities of the nodes and links. To do so, a REST-based interface is employed. In particular, once the TSN controller is contacted with a Get Topology request, it first consults its internal inventory. Then, it transforms the multiple representations of the elements into a useable Topology object, also augmenting the attributes of the multiple elements so as to have a richer and more descriptive model.

For the sake of illustrative purposes, let us consider that the workflow in Figure 2 is triggered by the provisioning request of flow TS_1 in the example of Figure 1b. Figure 3 depicts an extract of the Controller log, showcasing the reception of the petition coming from the CM and the gathering of the inventory by the controller. We also showcase the communication exchanges between the two components (label 1 of Figure 2) and the Topology object in JSON format that is returned in Figure 4. Once this information is received, the CM is ready to use the topology representation for path computation and end-to-end flow provisioning. Figure 5 shows the ENP Graphical User Interface (GUI) of the CM displaying the topology, including the network elements of the TSN domain and the calculated path for the TSN flow provision after the whole process.

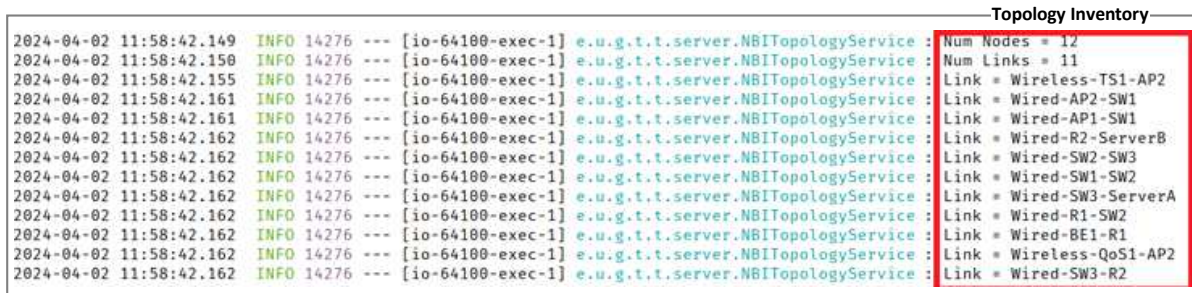


Figure 3: Node and link inventory at the TSN Controller

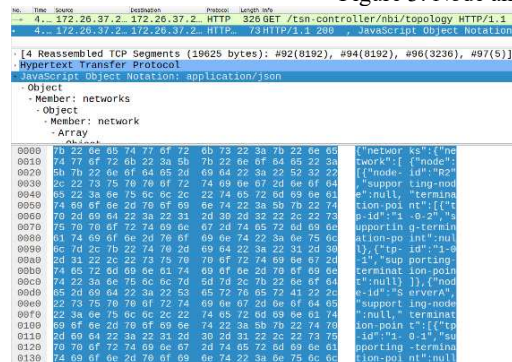


Figure 4: Topology and capabilities exposure

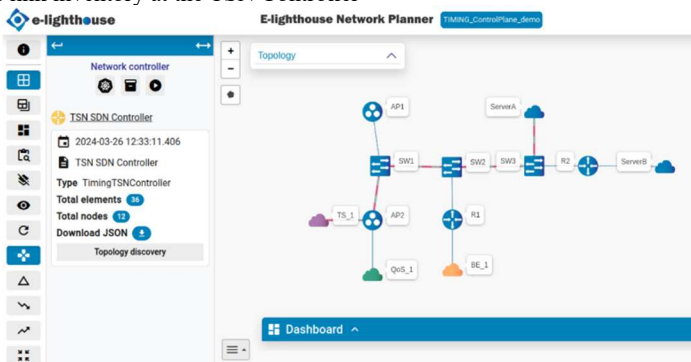


Figure 5: Topology in the CM GUI (ENP Front-End)

The DT, which is based on the architecture presented in [5] for generic AI-based purposes, is implemented in Python 3.10.4 and runs on a VM with Ubuntu Server 22.04 LTS as operating system. It contains the following basic modules: *i*) a manager module configuring and supervising the operation of the rest of the modules; *ii*) a number of modules that include algorithms, models, and the interface with the CM; and *iii*) a Redis DB that is used in publish-subscribe mode to communicate the different modules among them.

The DT builds a partition of the real network scenario defined by the path of the requested flow. This partition is emulated using basic components: *i*) *generators*, to produce synthetic flow traffic according to the statistical characteristics of each service and periodical (daily) profiles; *ii*) *flow queue models* based on the time-dependent ones presented in [6] and used to emulate TS-capable elements with different priority queues; *iii*) *links* to emulate the transmission delay of network links; and *iv*) *sinks*, used as end-points of flows for KPI evaluation purposes.

Request under evaluation	AP2	SW1	SW2	SW3	Server A
TS_1					
G	TSN	TSN	TSN	TSN	S
QoS ₁	QoS	QoS	QoS	QoS	
G	BE	BE	BE	BE	
Existing flows		G			Server B
	BE ₁				S

Figure 6: DT queuing model example

Figure 6 sketches the emulated partition that is run in the DT in the event of TS₁ flow provisioning. Recall that the DT is in charge of estimating the KPIs of requested and already existing non-TS flows. The propagation of the generated traffic for the flows through the defined queuing system results in metrics, such as queued traffic, that are afterwards used to compose flow KPIs, such as e2e delay. The DT computes two types of KPIs estimation: *i*) *e2e*, which are provided only for non-TS requests under evaluation; and *ii*) *partial*, computed for already deployed non-TS flows as the KPI variation (increment or decrement) for each flow if the new request would be finally deployed, considering the network partition defined by the request path. Figure 7 shows the content message that is sent from DT to CM (label 6 in Figure 2) once KPIs estimation is performed using the model in Figure 6. Since TS₁ is a TS flow, only partial KPIs of the existing non-TS flows are provided. In particular, for each non-TS flow and each evaluated day time, estimations of delay and jitter statistics (min, average, and max) are provided.

The demo will showcase the execution of the provisioning workflow for two different requests. On the one hand, the TS flow request TS₁ already presented in this section as illustrative example will be firstly presented. In this case, the new request will be provisioned since it will not affect the KPIs of established flows. On the other hand, and after provisioning TS₁ request, a new non-TS request with committed QoS (QoS₂) will be evaluated. The new topology to be evaluated is sketched in Figure 8a and requires extending the previous emulated partition with the new AP2. For the sake of visualization purposes, Figure 8b depicts the results of KPI estimation to be obtained after DT execution. In particular, the estimated maximum delay as a function of day time is presented for both new non-TS request and existing non-TS flows. In this case, provisioning QoS₂ flow request would seriously affect the performance of QoS flows, which would lead the CM to decide rejecting the path for the new request.

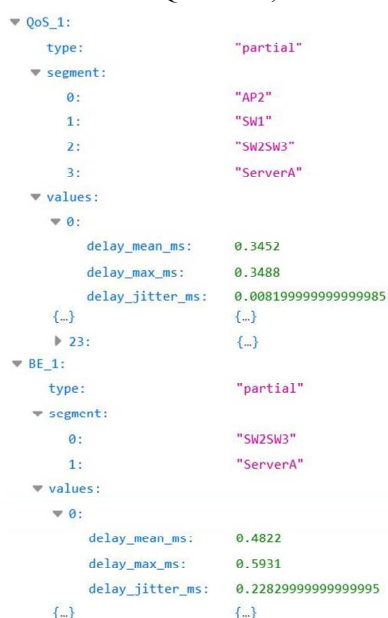


Figure 7: KPIs reply message

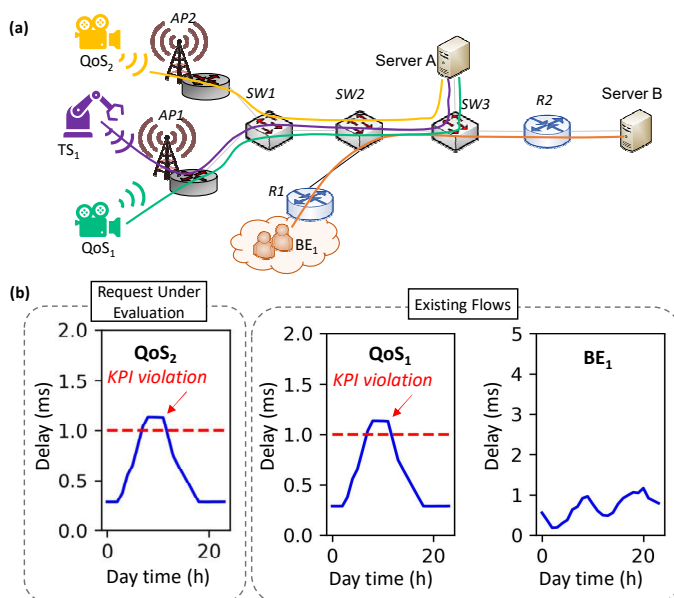


Figure 8: Example of KPI requirement violation

ACKNOWLEDGEMENTS

This research has received funding from the NextGeneration UNICO5G TIMING (TSI-063000-2021-145 and TSI-063000-2021-148), the SNS JU through the European Union's Horizon Europe under G.A. No. 101095890 (PREDICT-6G), the AEI IBON (PID2020-114135RB-I00) and TRAINER (PID2020-118011GB-C22) projects, and the ICREA institution.

REFERENCES

- [1] L. Velasco *et al.*, "End-to-End Intent-Based Networking," IEEE Communications Magazine, vol. 59, pp. 106-112, 2021.
- [2] IEEE Time-Sensitive Networking Task Group. [On-line] <https://1.ieee802.org/tsn/>.
- [3] A. Bernal *et al.*, "Near real-time estimation of end-to-end performance in converged fixed-mobile networks," Comp. Comms., vol. 150, 2020.
- [4] E-Lighthouse Network Planner. [On-line] <https://e-lighthouse.com/products/networkplanner>.
- [5] L. Velasco *et al.*, "Distributed Intelligence for Pervasive Optical Network Telemetry," IEEE/OPTICA JOCN, 2023.
- [6] L. Velasco and M. Ruiz, "Supporting time-sensitive and best-effort traffic on a common metro infrastructure," Comm. Letters, vol. 24, 2020.