Experimental Assessment of a Flow Controller for Dynamic Metro-Core Predictive Traffic Models Estimation

F. Morales[1*], Ll. Gifre[2], F. Paolucci[3], M. Ruiz[1], F. Cugini[4], L. Velasco[1] and P. Castoldi[3]

(1) Universitat Politècnica de Catalunya (UPC), Barcelona (Spain), fmorales@ac.upc.edu
(2) Universidad Autónoma de Madrid (UAM), Madrid (Spain)
(3) Scuola Superiore Sant'Anna, Pisa (Italy) (4) CNIT, Pisa (Italy)

Abstract A Flow Controller is proposed and experimentally assessed to share updated metro-flow predictive traffic models among metro and core controllers. The proposed controller allows a fast core flow traffic models re-estimation after flow traffic re-routing in metro areas.

Introduction

Many network operators are deploying differentiated metro and core segments, where metro areas are connected to the core through two or more nodes. In this network architecture, independent software-defined networking (SDN) controllers for the different segments (metro and core) and technologies (MPLS and optical) are usually deployed. However, this domain-managed network scenario might lead to local optimal resource allocation since flows in metro-areas (metro-flows) are routed considering only resource availability in the metro. In addition, the MPLS-over-optical Virtual Network Topology (VNT) in the core is usually designed assuming a traffic matrix built based on the metro flows entering that network. Therefore, the core VNT might become congested if metro-flows are re-routed and enter in the core through a different node because of metro-scope re-optimization.

To support and automate VNT adaptability, authors have proposed a data analytics enabled network manager architecture to store origin-destination (OD) traffic monitoring data featuring prediction[1] and detection of traffic anomalies[2]. Nonetheless, OD traffic patterns in the core might change as a result of metro-flow re-routing thus, making their predictive models obsolete.

A recent work[3] proposes to aggregate metro-flow predictive models estimated in metro areas to re-estimate obsolete OD models in the core based on statistical algorithms for predictive model aggregation. This allows keeping the predictive capabilities in the core uninterruptedly under metro-flow re-routing. Despite the savings obtained, the lack of network architectures supporting this multi-domain data analytics solution prevents its application in practice.

To solve this problem, we propose a Flow Controller where data regarding metro-flows is stored and can be queried by the controllers.

Traffic Model based on Model Aggregation

Although core OD traffic can be modelled by monitoring the aggregated traffic in the origin core node, that traffic model would become obsolete in case a metro controller decides to re-route some of the metro-flows in its metro area changing their ingress core node.

An example is shown in Fig. 1, where a core VNT interconnects metro networks using dual-homing. A metro-flow (*mf1*) originated at some area is routed towards datacenter DC2 in a different metro area through the core VNT. Initially, *mf1* enters the core VNT through ingress node R1 towards egress node R3, so it is part of OD R1->R3 traffic. Due to metro re-optimization, the metro controller decides to re-route *mf1*, so that it now enters the core VNT through core node R2, being aggregated into OD R2->R3. As a result, the traffic profiles of OD R1->R3 and R2->R3 have now changed.

Fig. 2 shows the traffic and predictive model of OD pair R1->R3. Before the re-routing event, the predictive model based on aggregated core traffic perfectly fits the real traffic; however, the re-routing of *mf1* changes the OD traffic profile thus making the predictive model obsolete, triggering a model re-estimation which would take several days or even weeks. Note also that the difference between actual traffic volume and the obsolete model traffic prediction can be

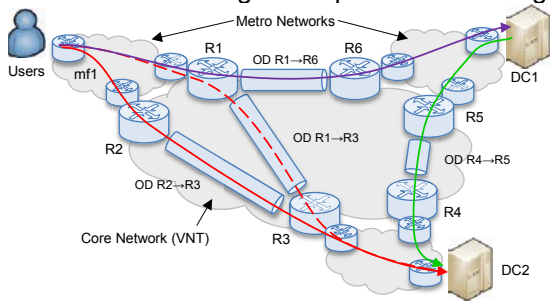

Fig. 1. OD traffic change after metro re-configuration.

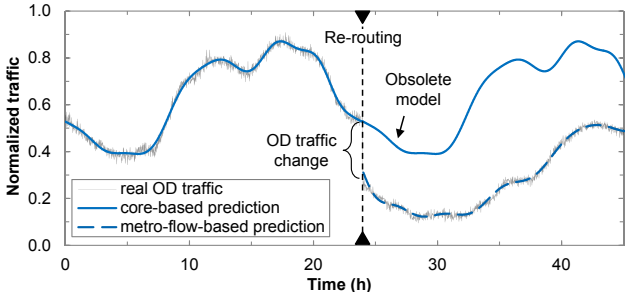

Fig. 2. Example of a traffic model that becomes obsolete.

978-1-5386-5624-2/17/$31.00 ©2017 IEEE

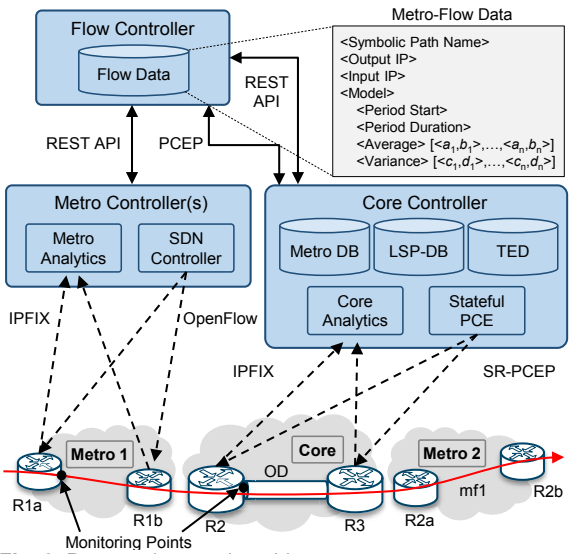

Fig. 3. Proposed network architecture.

mistakenly confused with an OD traffic anomaly, which would trigger unnecessary network reconfiguration[2].

Alternatively, OD traffic can be predicted by aggregating the metro-flow predictive models of the metro-flows being routed through each OD[3]. This, however, requires implementing some sort of coordination between metro and core segments, which we present next.

Proposed architecture and workflows

Fig. 3 presents the proposed architecture, where every metro controller contains a data analytics module[4] capable of storing and processing metro-flow monitored data to estimate traffic predictive models. The metro controller also includes an SDN controller responsible for the configuration of the network devices. The core controller contains an analogous analytics module for OD traffic monitoring and prediction, including an additional database to store the metro-flow -related data. A Traffic Engineering Database (TED), a Label Switched Path Database (LSP-DB), and a Stateful Path Computation Element (PCE) complete the core controller's architecture.

Finally, we propose a centralized Flow Controller with a repository to store flow-related data that is updated from the metro controllers and used by the core controller to produce predictive OD traffic models. The repository stores for each metro flow: *i*) its LSP's symbolic path name; *ii*) the border metro nodes through which the flow leaves and enters different metro areas. For instance, the output and input metro nodes for metro-flow *mf1* in Fig. 2 are R1b and R2a, respectively; and *iii*) the metro-flow predictive model. Since periodic models are assumed, the model includes its period and starting time, as well as two piecewise linear functions defining the average bitrate and its variance along the period.

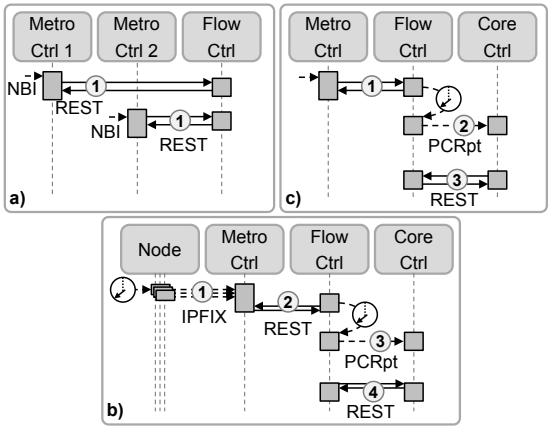

Fig. 4. Metro-flow set-up (a), Metro-flow model estimation (b) and Metro-flow re-routing (c).

Several workflows are defined to store and retrieve metro-flow data in/from the Flow Controller. Fig. 4a shows the first workflow triggered when a new LSP for a metro-flow is set-up across different metro domains. Every metro-controller involved in the LSP creation sends a JSON-encoded REST API message to the Flow Controller with the LSP symbolic path name and the input or output metro border node. The Flow Controller creates a new entry in the flow repository and populates it correlating the data received from different metro controllers.

Once the LSP for the metro-flow is operational, its traffic is monitored in any of the nodes in the source metro area (extended with monitoring and data analytics capabilities[4]) and monitored data is exported by means of IPFIX messages (message 1 in Fig. 4b) to the metro controller. When enough monitoring data has been collected to estimate a new metro-flow model, or to re-estimate an existing one, the metro controller sends the predictive model to the Flow Controller in a REST API message, including the LSP's symbolic path name for identification (message 2).

After a metro-flow data entry has been completed or updated, the Flow Controller notifies that to the core controller by issuing a PCEP PCReport message[5] (message 3) containing the list of updated metro-flow LSPs (a delay has been introduced to prevent flooding the core controller with several updates from different metro controllers). The core controller can now retrieve updated metro-flow data by issuing a REST API request (message 4), being thus able to update obsolete OD traffic models. When a metro controller decides to re-route one or more metro-flows, it updates the metro-flow data in the Flow Controller (message 1 in Fig. 4c), which proceeds as described for Fig. 4b.

Experimental assessment

Experiments to assess the proposed architecture have been carried out in a distributed test-bed connecting CNIT (Pisa, Italy) and UPC

(Barcelona, Spain) premises. A core controller with segment-routing capabilities[6] was located at CNIT, whereas the Flow Controller and two metro controllers were located at UPC. The core and metro controllers were extended with an HTTP REST API interface to exchange JSON-encoded messages with the Flow Controller. The SDN controller used in the metro areas is based on RyuSDN and uses OpenFlow to configure the network nodes. Finally, a set of extended nodes[4] implemented on top of OpenVSwitches were deployed using Mininet at UPC premises to allow monitoring traffic.

Fig. 5a lists the REST API messages exchanged between the metro controllers and Flow Controller after a LSP for a metro-flow across the two metro domains is set-up. Messages specify the LSP's symbolic path name (LSP-01-02) and the IP address of the metro border node. Fig. 5b shows an IPFIX flow message (labeled as 1 in Fig. 4b) containing monitoring data from LSP-01-02 that is sent to the source metro controller for traffic model estimation. After collecting enough traffic data, a predictive model is estimated by the metro controller and sent to the Flow Controller in a JSON-encoded REST API message (message 2). The details are shown in Fig. 6a and include the LSP symbolic path name and the data representing the metro-flow predictive model.

Next, the Flow Controller issues a PCEP PCReport message to the core controller notifying the new data available for LSP-01-02 (message 3). The PCReport message contains a list of Stateful Request Parameters (SRP) and one LSP object with the LSP's symbolic path name. The core controller then issues a REST-API request with the symbolic path name of the LSP (message 4) to retrieve its data (Fig. 6b).

Finally, Fig. 5c lists the messages exchanged as a result of re-routing of LSP-01-02. First, the new border output node is sent by the metro controller in a REST API message to the Flow Controller (labeled as 1; in Fig. 4c). Once the Flow Controller updates the metro-flow data, equivalent messages to those for model creation are exchanged with the core controller to allow obtaining updated data for the re-routed LSPs (notice the updated border node in Fig. 6c).

a)

Src	Dst	Info
MetroCtrl1	FlowCtrl	POST /flows?symPathName=LSP-01-02 HTTP/1.1
FlowCtrl	MetroCtrl1	HTTP/1.1 200 OK (application/json)
MetroCtrl2	FlowCtrl	POST /flows?symPathName=LSP-01-02 HTTP/1.1
FlowCtrl	MetroCtrl2	HTTP/1.1 200 OK (application/json)

b)

	Time	Src	Dst	Info
1	*REF*	10.0.10.11	MetroCtrl1	IPFIX flow (64 bytes) Obs-Domain-ID=258
2	0.173	MetroCtrl1	FlowCtrl	POST /flows?symPathName=LSP-01-02 HTTP/1.1
	0.189	FlowCtrl	MetroCtrl1	HTTP/1.1 200 OK (application/json)
3	60.476	FlowCtrl	CoreCtrl	Path Computation LSP State Report (PCRpt)
4	60.521	CoreCtrl	FlowCtrl	GET /flows?symPathName=LSP-01-02 HTTP/1.1
	60.529	FlowCtrl	CoreCtrl	HTTP/1.1 200 OK (application/json)

c)

	Src	Dst	Info
1	MetroCtrl1	FlowCtrl	POST /flows?symPathName=LSP-01-02 HTTP/1.1
	FlowCtrl	MetroCtrl1	HTTP/1.1 200 OK (application/json)
2	FlowCtrl	CoreCtrl	Path Computation LSP State Report (PCRpt)
3	CoreCtrl	FlowCtrl	GET /flows?symPathName=LSP-01-02 HTTP/1.1
	FlowCtrl	CoreCtrl	HTTP/1.1 200 OK (application/json)

Fig. 5. Messages list for metro-flow set-up (a), metro-flow traffic model update (b) and metro-flow LSP re-routing (c).

Conclusions

A Flow Controller has been proposed to allow metro controllers to share metro-flow predictive traffic models with the core controller. This facilitates the core controller re-estimating obsolete OD traffic models applying statistics for metro-flow model aggregation.

Three workflows have been proposed to keep updated metro-flow data in the Flow Controller, either triggered by the estimation of a new predictive traffic model or for a re-routing action modifying the OD traffic aggregation in the core network. In any case, these actions originated by metro controllers are properly notified to the Flow Controller and eventually to the core controller.

The architecture has been experimentally assessed in a distributed test-bed connecting CNIT and UPC premises.

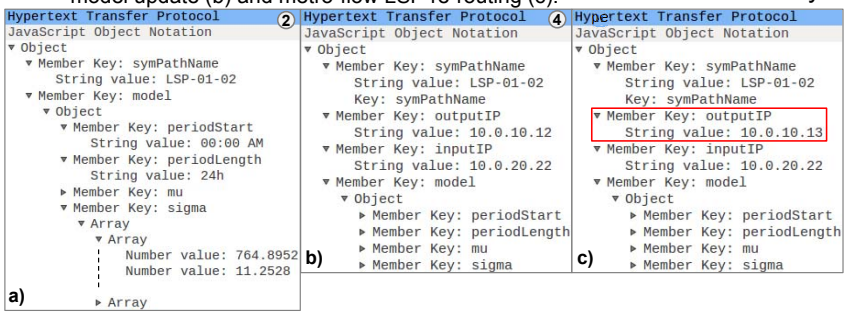

Fig. 6. Details of traffic model (a), metro-flow (b) and updated metro-flow data (c).

Acknowledgements

This work was partially supported by the EC through the ORCHESTRA (G.A. nº 645360) and METRO-HAUL (G.A. 761727) projects, from the Spanish MINECO SYNERGY project (TEC2014-59995-R) and from the Catalan Institution for Research and Advanced Studies (ICREA).

References

1 F. Morales et al., "Virtual Network Topology Adaptability based on Data Analytics for Traffic Prediction," IEEE/OSA JOCN vol. 9, pp. A35-A45, 2017.

2 A. P. Vela et al., "Distributing Data Analytics for Efficient Multiple Traffic Anomalies Detection," Computer Communications, vol. 107, pp. 1-12, 2017.

3 F. Morales et al., "Core VNT Adaptation Based on the Aggregated Metro-Flow Traffic Model Prediction", in Proc. OFC 2017.

4 Ll. Gifre et al., "Experimental Assessment of Node and Control Architecture to Support the Observe-Analyze-Act Loop," in Proc. OFC 2017.

5 E. Crabbe et al., "PCEP Extensions for Stateful PCE," IETF draft, work in progress, 2016.

6 A. Sgambelluri et al., "Experimental Demonstration of Segment Routing," IEEE/OSA JLT, vol. 34, pp. 205-212, 2016.