Big Data Analytics in Support of Virtual Network Topology Adaptability

Lluís Gifre[1], Luis. M. Contreras[2], Victor López[2], and Luis Velasco[1*]

Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
[2] Telefónica Investigación y Desarrollo (TID), Madrid, Spain.
**e-mail:lvelasco@ac.upc.edu*

Abstract: ABNO's OAM Handler is extended with big data analytics capabilities to anticipate traffic changes in volume and direction. Predicted traffic is used as input for VNT re-optimization. Experimental assessment is realized on UPC's SYNERGY testbed.

OCIS codes: (060.4250) Networks; (060.4256) Networks, network optimization

1. Introduction

Static network topologies, where large packet nodes (e.g., IP routers) connected through virtual links (*vlinks*) supported by static connections in the optical layer, were commonly designed to cope with the traffic forecast. However, the introduction of new type of services requiring large bitrate connectivity (e.g., datacenter interconnection, CDN for live-TV and video distribution, etc.) together with the traffic increment that operators' networks are needing to deal with year after year entails that static packet network topologies were largely overprovisioned thus, increasing network total cost of ownership (TCO). In view of that, network operators are looking for more efficient architectures able to reduce TCO, while providing the required grade of service. To that end, virtual network topologies (VNT) need to be dynamically adapted not only to variations in traffic volume, but also to changes in the direction of the traffic.

To support connectivity dynamicity, the IETF has recently standardized the ABNO architecture [1], which includes among others: *i*) the ABNO controller as the entrance point to the network for provisioning and advanced network coordination. It acts as a system orchestrator invoking its inner components according to a specific workflow; *ii*) a virtual network topology manager (VNTM) in charge of reconfiguring the on demand VNT; *iii*) a Path Computation Element (PCE) to compute the path for new label switched paths (LSP); *iv*) a provisioning manager (e.g., a SDN controller) responsible for managing LSPs, both at the optical layer (Lambda Switch Capable, LSC) and at the packet layer (Packet Switch Capable PSC); and *v*) the Operations, Administration, and Maintenance (OAM) Handler to receive notifications and monitored counters.

To automate VNT adaptability, traffic needs to be monitored in the packet nodes and counters be accessible by the OAM Handler. In particular, the disaggregated traffic volume forwarded by each packet node to every other destination node should be available. In addition, notification can be also triggered when the used vlink capacity reach some configured threshold (e.g., 90%). In fact, threshold triggered VNT reconfiguration where the OAM Handler receives notifications from the control plane and reroutes individual PSC LSPs was proposed in [2]. Nonetheless, VNT adaptation requires from powerful architectures and algorithms to analyze large amounts of monitored traffic data, so as to anticipate, when possible, to traffic changes targeting at optimizing resource utilization. In this paper, we propose a big data network manager architecture to support VNT adaptability based on traffic prediction from applying data analytics on the monitored traffic data.

2. Threshold triggered vs. data analytics -based VNT reconfiguration

A threshold triggered VNT reconfiguration is shown in Fig. 1 for a seven-node VNT. Fig. 1(a) presents monitored traffic data captured during the last two hours in node 6, where traffic from that node to every other node in the VNT (6->*N*), from node 6 to node 7 (6->7), and from node 6 to every other node except to node 7 (6->*N*\{7}) is plotted. Fig. 1(b) shows the initial VNT where every vlink is supported by a 100 Gb/s LSC LSP in the underlying optical layer; the route of PSC LSP 6-7 is also shown. A 90% threshold is configured and, in the event of threshold violation, the OAM Handler might request the ABNO controller increasing the capacity of some vlinks by establishing a parallel LSC LSP. In our example, two threshold violations for vlinks 1-6 and 1-7 are received, so the VNT is updated (Fig. 1(c)). It is worth noting that no PSC LSP is rerouted.

Threshold triggered VNT reconfiguration is able to adapt the VNT to traffic changes at the cost of using an increased number of transponders in the nodes. In the example in Fig. 1, four optical transponders are used for the new LSC LSPs 1-6 and 1-7. However, analyzing the plots in Fig. 1(a) we realize that traffic 6->7 is responsible for the registered traffic increment. To facilitate data analysis, monitored traffic data received in the OAM Handler is aggregated into four values (i.e., min, max, average, and last) (see algorithm in Table 1) so as to model the traffic behavior during the last hour (Fig. 2 (a)); the algorithm follows a map-reduce scheme, where the map phase de-serializes JSON messages and the reduce phase merges sample pairs.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 317999 IDEALIST project and from the Spanish MINECO SYNERGY project (TEC2014-59995-R).

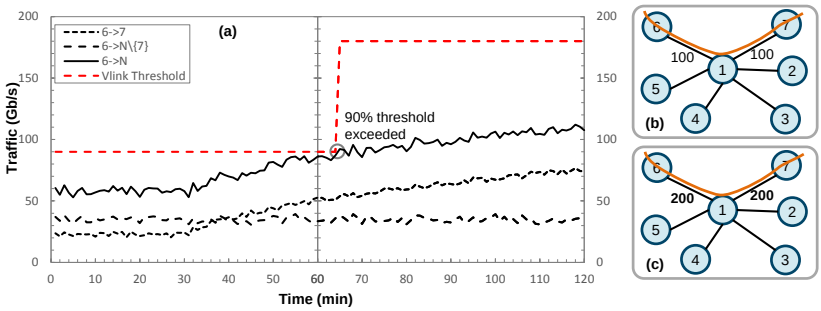

Fig. 1. Threshold triggered VNT reconfiguration.

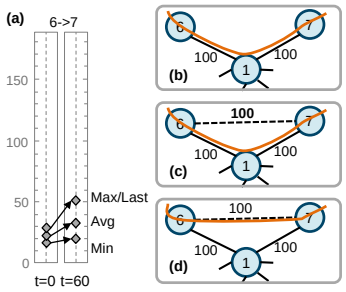

Fig. 2. Data analytics -based VNT reconfig.

From the current VNT (Fig. 2 (b)) and analyzing modelled traffic data, it can be seen that maximum and last traffics got similar values in *t*=60, being both over the average one. If traffic volume is large enough, vlink 6-7 can be created by establishing a LSC LSP (Fig. 2 (c)) and traffic 6->7 be non-disruptively rerouted (Fig. 2 (d)). Note that this solution needs from just two transponders in contrast to the threshold triggered reconfiguration. If the capacity of some vlinks can be reduced after rerouting, some of the supporting LSC LSPs will be torn-down.

Table 1. Aggregation Algorithm

INPUT T = {Text}
OUTPUT q = <*t, min, max, avg, last*>

```
1:  define aggregator(in p1, p2; out q1)
2:      q1.min ← min(p1.min, p2.min)
3:      q1.max ← max(p1.max, p2.max)
4:      q1.avg ← p1.avg + p2.avg
5:      q1.last ← getLastAvg(p1, p2)
6:      q1.t ← max(p1.t, p2.t)
7:      return q1
8:  P = {<t, min, max, avg>} ←
            T.map (jsonDeserialize)
9:  q ← P.reduce (aggregator)
10: q.avg ← q.avg / |P|
11: return q
```

3. Big data network manager: architecture and workflow

In the proposed architecture in Fig. 3, monitored traffic data is sent periodically to a big data repository consisting of a distributed database and a data collector. Periodically, e.g., every minute, the OAM Handler retrieves aggregated monitored data, which is stored into a big data system ready to be analyzed. The big data OAM Handler applies data stream mining techniques on the received data and periodically (e.g., every hour) transforms monitored data into modelled data. Modelled data is used by a *prediction* module, running machine learning algorithms to anticipate next period traffic conditions (labelled as 1 in Fig. 4). Based on the predicted traffic, a *decision maker* module decides whether the VNT should be updated. In case of VNT reconfiguration, the VNTM is in charge of computing the new VNT. To that end, the OAM Handler issues a request to the ABNO controller that includes the predicted traffic together with some other parameters to facilitate VNTM computation (2) and the ABNO controller initiates the workflow forwarding a request to the VNTM (3).

The VNTM computes the optimal VNT with the predicted traffic matrix received from the OAM Handler (4). Continuing with our seven-node VNT example, let us assume that the new VNT consists on adding the new virtual link 6-7 and reducing the capacity of some other vlinks. The solution is first notified to the NMS (5) and then, its implementation is divided into a sequence to avoid traffic disruption as anticipated above: firstly, LSC LSP 6-7 is created (6) and the new vlink is advertised (7); next, PSC LSPs are rerouted (8) (a make-before-brake strategy to avoid disruption can be implemented) and unused capacity in vlinks 1-3 and 1-4 removed by tearing down the underlying LSC LSPs (9); new vlinks' capacity is advertised (10). Upon VNT reconfiguration completion, VNTM replies the ABNO controller (11), which eventually replies the OAM Handler (12).

4. Experimental assessment

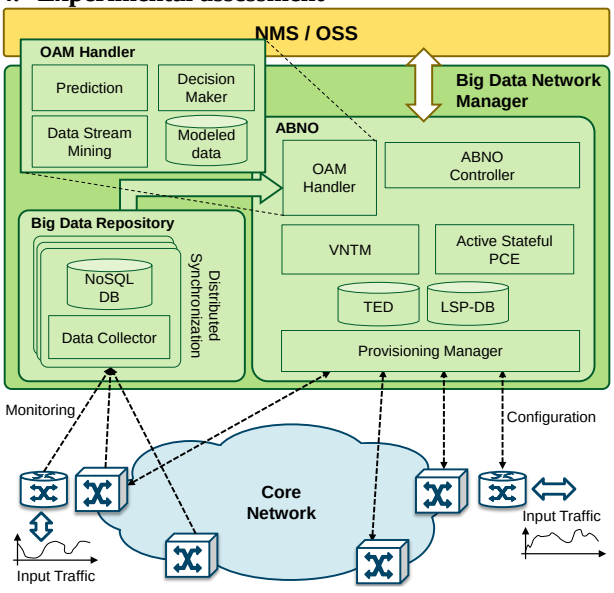

Fig. 3. Big data network manager architecture.

Experiments have been carried out on the UPC's SYNERGY test-bed. Apache Cassandra database [3] was used as a big data repository and the data collector module was implemented to offer an UDP-

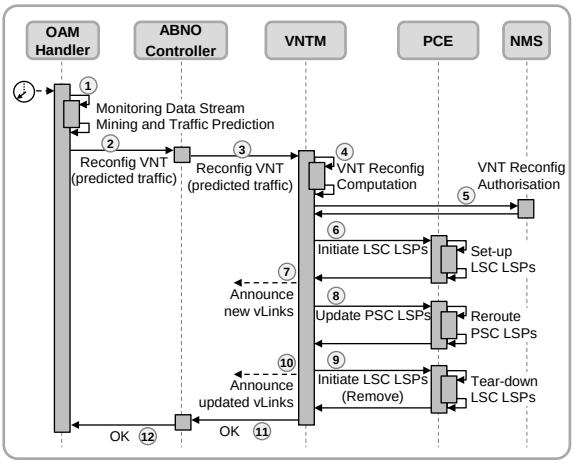

Fig. 4. Proposed workflow.

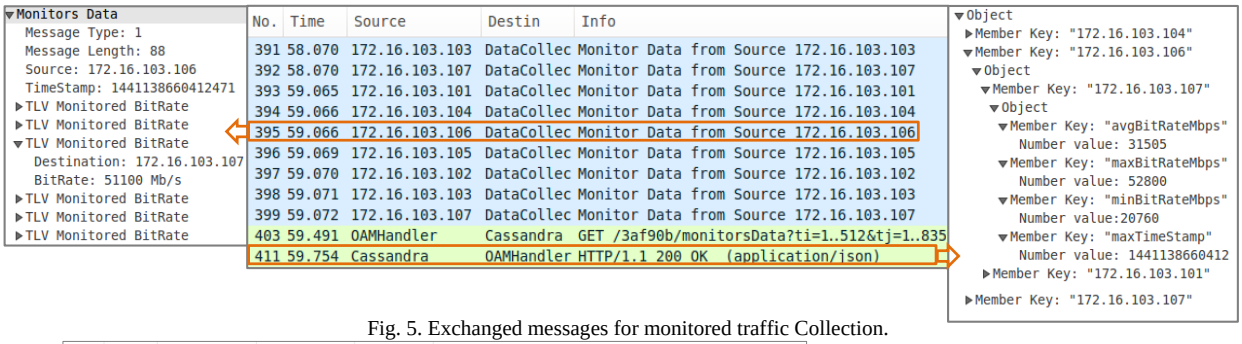

Fig. 5. Exchanged messages for monitored traffic Collection.

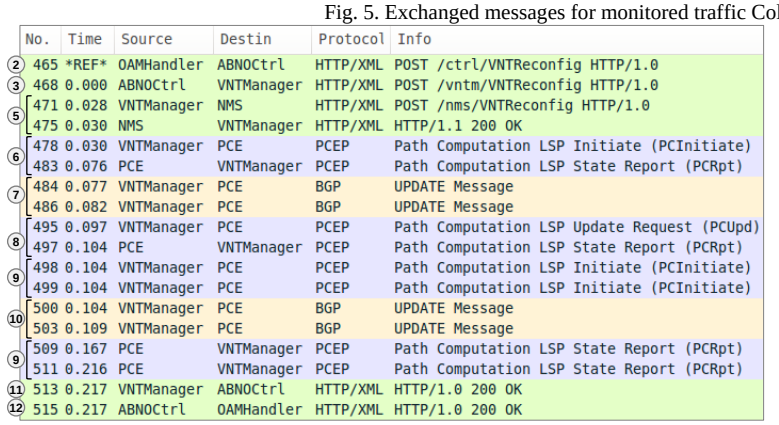

Fig. 6. Exchanged messages for VNT reconfiguration.

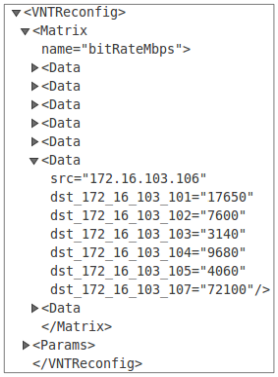

Fig. 7. Message (2) details.

based interface to the monitors, storing the received data in Cassandra. Apache Spark [4] was used to implement data stream mining and machine learning techniques. Finally, ABNO modules in Fig. 3 were implemented using UPC's iONE software [5]. A HTTP REST API interface was implemented between the OAM Handler and the ABNO controller and from it to VNTM, so as to convey the predicted traffic matrix. PCEP was used between VNTM, PCE, and Provision Manager. Finally, BGP-LS was used to synchronize traffic engineering databases (TED). In particular, VNTM is in charge of advertising topological changes in the VNT, including vlink creation and releasing, as well as updating vlink capacity changes.

Fig. 5 illustrates monitored traffic data periodically being send by the packet nodes to the data collector, as well as the request that the OAM Handler issues to Cassandra's REST API to collect monitored data. UDP monitoring messages contain, among others, the source node and the timestamp of the sample, and for each aggregated flow leaving the node to a destination, its destination node and bitrate. After selecting and aggregating monitored data between the selected times t_i and t_j, Cassandra replies with a JSON-encoded matrix specifying for each pair of source-destination the average, maximum, and minimum bitrate.

Fig. 6 shows the meaningful messages exchanged between ABNO modules. For the sake of clarity, messages are identified following the workflow in Fig. 4. The OAM handler sends a REST API request to the ABNO controller (message 2) containing the predicted traffic matrix for the next period. The details of that message are presented in Fig. 7. After receiving the predicted traffic matrix, the VNT computes the optimal VNT and issues requests to the PCE to implement the LSC LSPs supporting the new vlinks, reroute the selected PSC LSPs, and tear down unused LSC LSPs. In addition, VNT changes are advertised to the rest of ABNO modules. The total process took 217ms, from the instant the OAM handler triggered the workflow.

5. Conclusions

A big data analytics OAM handler has been proposed to support VNT adaptability based on traffic prediction. Packet nodes monitor incoming traffic and send monitoring data to a big data repository, based on Cassandra. Monitoring data is collected by the OAM Handler and locally stored. Periodically, e.g., every hour, collected monitoring data is transformed into modelled data and used to predict next period traffic applying machine learning techniques.

A workflow is proposed, where the VNTM module is in charge of adapting the VNT to future conditions. To that end, the VNTM finds the optimal VNT based on the predicted traffic computed by the OAM handler.

The architecture has been experimentally assessed in UPC's SYNERGY test-bed, where Apache Spark was used as a platform for data stream mining and machine learning purposes.

References

[1] D. King and A. Farrel, "A PCE-based Architecture for Application-based Network Operations," IETF RFC 7491, 2015.
[2] A. Aguado et al., "Dynamic VN Reconfiguration over SDN Orchestrated Multi-Technology Optical Domains," in Proc. ECOC 2015.
[3] Apache Cassandra: http://cassandra.apache.org/
[4] Apache Spark: http://spark.apache.org/
[5] L. Velasco and Ll. Gifre, "iONE: A Workflow-Oriented ABNO Implementation," in Proc. Photonics in Switching, 2015.