Reducing Virtual Network Reconfiguration and Traffic Losses under Multiple Traffic Anomalies

Alba P. Vela*, Marc Ruiz, and Luis Velasco
Universitat Politècnica de Catalunya (UPC). Barcelona, Spain
**apvela@ac.upc.edu*

Abstract: Multiple OD traffic anomalies can arise exceeding pre-planned capacity and causing congestion. We propose to detect anomalous and suspicious ODs to reduce virtual network reconfiguration and total traffic losses. Results show savings higher than 25%.
OCIS codes: (060.4250) Networks; (060.4256) Networks, network optimization.

1. Introduction

Traffic anomalies are sudden and short-living traffic changes affecting volume or direction or both and that do not follow any expected pattern [1]. They can create congestion in the network causing traffic loss and therefore, its prompt detection becomes essential since it allows to proactively reconfigure the network, e.g. by increasing the capacity of virtual links in multi-layer MPLS over optical networks [2].

Traffic anomalies can affect one or more Origin-Destination (OD) pairs. When a traffic anomaly involves only one single OD pair $o \rightarrow d$, a virtual network reconfiguration can be immediately triggered to increase the capacity allocated for traffic between origin router $o \in R$ and destination router $d \in R$ (e.g., by setting up new connections on the underlying optical network), where R represents the set of routers in the network. However, multiple anomalies can be caused, e.g., in the event of disaster affects the network [3]. In such case, several ODs leaving from a common origin router ($o \rightarrow (M \subset R)$) or arriving to a common destination router ($(N \subset R) \rightarrow d$) can be produced, where M represents the set of destinations from o and N is the set of origins from d. In such cases, reconfiguring the virtual network after an individual OD traffic anomaly is detected can result in both, an intolerable number of virtual network reconfiguration and traffic losses and a no global optimal configuration in terms of resource utilization.

In our previous work [4], we evaluated the performance of different OD traffic anomaly detection methods for the case of single OD traffic anomalies; we concluded that OD anomaly detection should be performed directly by the routers. Nevertheless, in view of the above, it is clear the need to develop efficient techniques to be carried out in the *network controller* (or the Network Management System) to detect the probable occurrence of related anomalies. In consequence, this paper focuses in scenarios with multiple OD anomalies arising within short time intervals.

Particularly, we address the decision making process to be carried out after an OD anomaly has been detected. Two actions can be taken: *i)* to wait for the evidence of more anomalous ODs; *ii)* to reconfigure those ODs with enough evidence of being anomalous. We propose the Anomaly and Network Reconfiguration (ALCOR) method with the aim to anticipate OD anomalies detection after an anomaly in a single OD pair has been detected. To that end, we extend [4] and redefine the distance function as a *score* to express the likelihood of an OD to be anomalous.

2. Multiple Anomalies and Network Reconfiguration

Let us assume an architecture where data analytics are distributed between the network controller and the routers (Fig. 1a). The controller configures every router in the network specifying, among other parameters, the anomaly

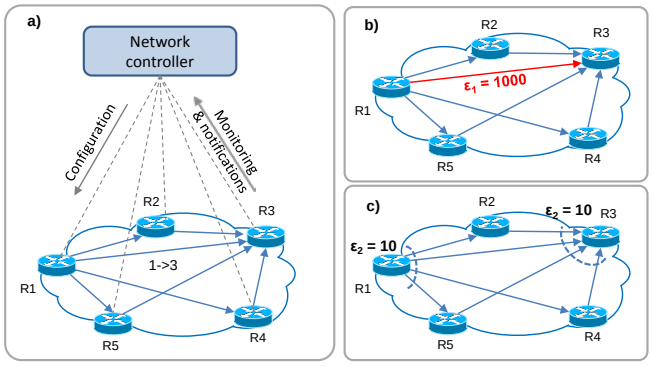

Fig. 1. a) Monitoring architecture. b) OD pair R1→R3 anomaly detection. c) Anomaly threshold notification reconfigured in R1 and R3.

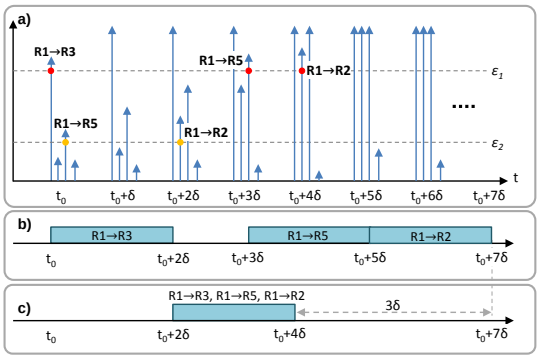

Fig. 2. Example of multiple anomalies. a) Score vs. time. b) Single and c) multiple OD reconfiguration.

The research leading to these results has received funding from the Spanish MINECO SYNERGY project (TEC2014-59995-R) and from the Catalan Institution for Research and Advanced Studies (ICREA).

detection thresholds (we use two different thresholds ε_1 to detect OD anomalies and ε_2 to detect suspicious OD pairs). Initially, routers only monitor outgoing OD traffic and periodically send average values towards the controller, which estimates the coefficients to model OD traffic. The coefficients for every OD pair are forwarded to the origin and destination routers and are used together with the anomaly thresholds to detect traffic anomalies. Whenever an OD traffic anomaly is detected by a router, a notification is sent to the controller.

Fig. 1b shows a particular example where a traffic anomaly has been detected in OD *R1→R3* (note that threshold ε_1 has been violated); the score evolution with time is shown in Fig. 2a. Note that a purely *per-OD* reconfiguration would immediately trigger a network reconfiguration after the *R1→R3* OD traffic anomaly is detected (Fig. 2b). Considering that the reconfiguration process takes δ time units (e.g., 2 minutes), this process will end at $t_0+2\delta$. Later, at $t_0+3\delta$, an OD anomaly in OD pair *R1→R5* is confirmed and thus, the controller triggers a new reconfiguration for such OD that finishes at $t_0+5\delta$. While the reconfiguration for OD pair *R1→R5* is still in progress, another anomaly for OD pair *R1→R2* is detected at $t_0+4\delta$, but the reconfiguration for that OD anomaly needs to be delayed until that for *R1→R5* OD pair finishes. Therefore, reconfiguration for *R1→R5* OD pair starts at $t_0+5\delta$ and finishes at $t_0+7\delta$.

To reduce the number of reconfigurations and the time the last one finishes thus, reducing traffic losses, a procedure to detect multiple OD anomalies would be useful. To that end, we propose the ALCOR method to analyze other ODs; when the controller receives *R1→R3* OD traffic anomaly notification, it decides to activate the notification for threshold ε_2 in router *R1* for all outgoing OD pairs *R1→N*, as well as to activate traffic monitoring for the incoming ODs *M→R3* and ε_2 threshold notification in *R3* (Fig. 1c). When threshold ε_2 is enabled in router *R1* at time t_0, a threshold violation notification is immediately sent to the controller for *R1→R5* OD pair and two decisions can be made: *i*) trigger a network reconfiguration for ODs *R1→R3* and *R1→R5* (assuming a traffic anomaly for the latter); *ii*) wait for traffic anomaly evidence affecting more ODs. Assuming the latter, the reconfiguration is triggered in $t_0+2\delta$ for all three ODs and finishes at $t_0+4\delta$ (Fig. 2c) therefore, preparing the network for the new traffic conditions 3δ time units before than the per-OD reconfiguration.

3. The Anomaly and Network Reconfiguration (ALCOR) method

This section presents the ALCOR method that identifies multiple related anomalies by redefining the distance function of an OD pair in [4] to a value named as score $s(t)=1/\Pi_{i=0..|H|-1}\ P(x \geq |\ (p_{t-i} - \mu_{t-i}) / \sigma_{t-i}\ |,\ x \sim N(0,1))$, where p_{t-i} is the measured traffic value, and μ_{t-i} and σ_{t-i} are the expected mean and standard deviation at time *t-i*. We use the score to decide whether to trigger a network reconfiguration or to wait for further anomaly evidences. ALCOR decision problem can be addressed by solving the algorithm presented in Table 1. The algorithm receives an *od*∈*OD* in the set of OD pairs in the network, for which an anomaly threshold *thr* has been exceed at time *t*. Two different thresholds are considered: ε_1 for anomalous and ε_2 for suspicious OD pairs. The algorithm returns a set of tuples with the OD pairs to be reconfigured together with the threshold violated and the time of such event.

An ε_1 violation detected in a router *o* at time *t* for OD pair *od* triggers the ALCOR algorithm (Table 1) in the controller; auxiliary sets are initialized and variable *decisionOngoing* is set (lines 1-3 in Table 1).

Table 1. ALCOR algorithm.

```
INPUT: od, thr, t                    OUTPUT: Sol
 1:  if decisionOngoing=false then
 2:      Sol ←Ø; M2i ←Ø; M2o ←Ø
 3:      decisionOngoing = true
 4:  if thr= ε1 then
 5:      <M1, M2i, M2o>←
             reconfigureODMonitoring(M1, M2i, M2o, od)
 6:      Sol ←Sol ∪ {<od,thr,t>}
 7:      configureTimer (time, ALCOR(Ø, 0, 0))
 8:      return Ø
 9:  if od≠ Ø then
10:      Sol ←Sol ∪ {<od,thr,t>}
11:      removeTimer()
12:  if reconfigureNow(Sol) then
13:      removeMonitoring(M2i)
14:      configureMontoring(M2o, ε1)
15:      M1 ←M1 ∪ M2o
16:      decisionOngoing = false
17:      return Sol
18:  configureTimer (time, ALCOR(Ø, 0, 0))
19:  return Ø
```

Next, the network controller configures monitoring parameters to detect ε_2 violations for every other OD pair *od* leaving router *o* and entering router *d* (line 5). Note that initially only outgoing OD traffic was monitored to detect ε_1 violations. Sets of monitored ODs are updated, containing those ODs being monitored for threshold ε_1 (*M1*), outgoing ODs for threshold ε_2 (*M2o*) and incoming ODs for threshold ε_2 (*M2i*). The partial solution *Sol* is updated with the detected anomaly and a timer is started waiting for new evidences (lines 6-7). When the controller receives an ε_2 violation, *Sol* is updated and timers disabled (lines 9-11).

Function *reconfigureNow*(·) computes a trade-off between the benefit and risk of waiting for new evidences of triggering a network reconfiguration. If network reconfiguration is decided, the controller first reconfigures monitoring in the routers and returns those OD with enough evidences of being anomalous (those exceeding any threshold) (lines 12-17). Otherwise, a new timer is started waiting for more evidences.

Table 2. Anomaly scenario 1

	per-OD	ALCOR	Savings
Losses (>3σ)	10.6	5.4	49%
Reconfig (min)	8	4	50%
# Reconfigs	4	1	75%

Table 3. Anomaly scenario 2

	per-OD	ALCOR	Savings
Losses (>3σ)	6.2	4.5	27%
Reconfig (min)	9	9	0%
# Reconfigs	4	3	25%

4. Illustrative numerical results

For evaluation purposes we developed an ad-hoc event-driven simulator in OMNET++ to generate traffic [5] and anomalies on a ten-router network. Generated values were used as input of an R function that computed the score of every OD pair and returned whether an OD exceeded a threshold. Finally, the ALCOR algorithm was developed as an R function.

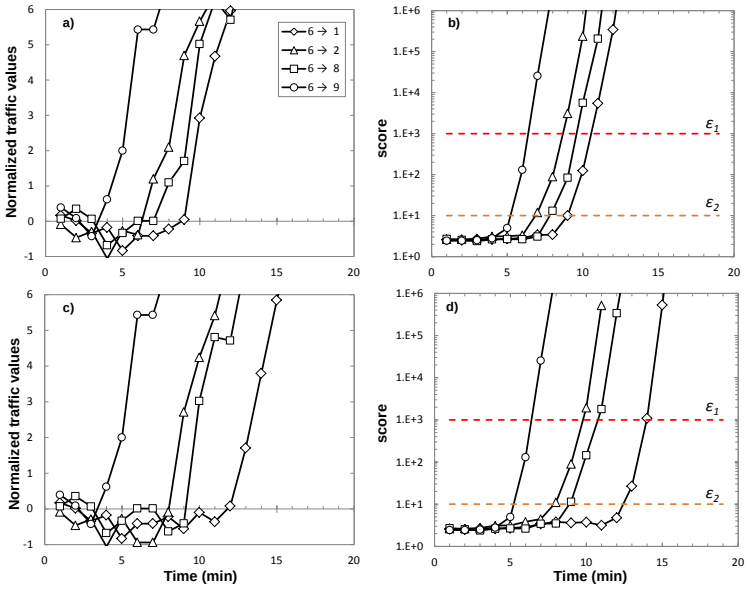

Fig. 3 Normalized traffic and score values against time for anomaly scenario 1 (a-b) and for anomaly scenario 2 (c-d).

Graphs in Fig. 3 presents the results for two different scenarios, when anomalies arrive close in time one to the other (scenario 1 in Fig. 3a-b) or more spaced (scenario 2 Fig. 3c-d).

Fig. 3a and Fig. 3c plot the evolution of normalized traffic values as a function of the time; by construction they should be centered on zero (the mean value) and within some interval in terms of σ. When a traffic anomaly occurs in an OD, the normalized traffic increases (or decreases) sharply; four anomalous ODs are thus, observed in the graphs. However, it is difficult to set thresholds for normalized traffic values since the probability of observing values out normal boundaries (e.g., 3σ) is not negligible. Therefore, the score presented in section 3 that considers previous traffic values is used; Fig. 3b and Fig. 3d plot the computed scores against time for the four ODs in the considered scenarios. Note the difference between normalized traffic and score for OD pairs 6→9 and 6→2 in scenario 1 (Fig. 3a). Although the normalized traffic value is around 2σ at minute 5 and minute 8 for OD pairs 6→9 and 6→2, respectively, their score values are under ε_2 for OD pair 6→9 and above ε_2 for OD pair 6→2.

Comparing both scenarios, it is clear that ALCOR would made different decisions in each case. For instance, in scenario 1, when the first anomaly in OD pair 6→9 is detected at minute 7, OD pair 6→2 has already exceeded ε_2 threshold. In the following two minutes, first OD pair 6→8 and then OD pair 6→1 also violate ε_2. In this scenario, ALCOR would detect such and wait until evidences of traffic anomaly in all four OD are detected and so, only one reconfiguration would be triggered at minute 9. Table 2 compares to the per-OD and ALCOR strategies. Interestingly, the per-OD reconfiguration would need 4 reconfigurations and delay the network adaptation in 4 minutes, doubling traffic losses with respect to ALCOR.

Under scenario 2, the first OD traffic anomaly is detected in pair 6→9 at minute 7. Since no evidences of other anomalous OD are found, ALCOR decides to trigger a reconfiguration for this OD. When another OD traffic anomaly is detected in pair 6→2 at minute 10, OD pair 6→8 also violates ε_2 and ALCOR triggers another reconfiguration for these ODs. Finally, a third reconfiguration is triggered for OD pair 6→1 at minute 14. In contrast, 4 reconfigurations would be triggered by the per-OD strategy and although the network would be adapted to the new conditions at the same time, the per-OD strategy increases traffic losses in a 27% (Table 3).

5. Conclusions

The ALCOR method has been proposed to identify multiple related OD traffic anomalies targeting at reducing traffic losses coming from unexpected traffic increments, as well as the number of network reconfigurations performed to consequently adapt the network capacity. Simulation results show remarkable saving compared to a per-OD reconfiguration strategy.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” ACM Computing Surveys, vol. 41, pp. 1-72, 2009.
[2] F. Morales et al, “Virtual Network Topology Reconfiguration based on Big Data Analytics for Traffic Prediction,” in Proc. OFC, 2016.
[3] Z. Nasralla et al., “Routing Post-Disaster Traffic Floods in Optical Core Networks”, in Proc. ONDM, 2016.
[4] A. P. Vela, A. Via, M. Ruiz, and L. Velasco, “Bringing Data Analytics to the Network Nodes,” accepted in ECOC, 2016.
[5] A. P. Vela, A. Via, F. Morales, M. Ruiz, and L. Velasco, “Traffic generation for telecom cloud -based simulation,” in Proc. ICTON, 2016.