Bringing Data Analytics to the Network Nodes for Efficient Traffic Anomalies Detection

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

ABSTRACT
Traffic anomalies can create network congestion, so its prompt and accurate detection would allow network operators to make decisions to guarantee the network performance avoiding services to experience any perturbation. In this paper, we focus on origin–destination (OD) traffic anomalies; to efficiently detect those, we study two different anomaly detection methods based on data analytics and combine them with three monitoring strategies. In view of the short monitoring period needed to reduce anomaly detection, which entails large amount of monitoring data to be collected and analyzed in a centralized repository, we propose bringing data analytics to the network nodes to efficiently detect traffic anomalies, while keeping traffic estimation centralized. Exhaustive simulation results on a realistic network scenario show that the monitoring period should be as low as possible (e.g., 1 min) to keep anomaly detection times low, which clearly motivates to place traffic anomaly detection function in the network nodes.
Keywords: data analytics, traffic anomalies.

1. INTRODUCTION

Traffic anomalies are short-living events that do not follow expected patterns (see a survey in [1]). They can create network congestion and stress resource utilization in packet nodes and hence, its prompt detection becomes essential since it allows preparing the network e.g., by reconfiguring the virtual network topology in multilayer network scenarios [2]. Anomaly detection can be used to trigger lightpath provisioning and network re-configuration when a traffic anomaly is detected [3], which entails analyzing monitoring data to anticipate traffic congestion. It is clear that developing efficient techniques to detect traffic anomalies in real time would empower network operators to prevent grave consequences induced by such anomalies affecting end users.

In order to detect traffic anomalies, it is essential to monitor traffic at the nodes and to model such traffic [4]. For packet networks specifically, the traffic monitoring function allows identifying (classifying) the traffic belonging to a specific service or destination, so as to apply specific policies. Monitoring traffic samples are produced at the packet nodes; according to the ITU-T [5], performance events are counted second by second over every 15-minute period. At the end of a period, they are collected in a repository for further analysis [2], [6]. It is clear that when analytics are applied to data collected every 15 minutes, the expected traffic anomaly detection times will be as well in that order of magnitude. Consequently, the monitoring period should be reduced, which in turn increases the amount of monitoring data to be sent to the centralized data repository.

In view of the above, several architectures need to be studied to evaluate different data analytics algorithms placements to reduce traffic anomaly detection time and the amount of monitoring data to be conveyed to the central repository.

2. OD TRAFFIC ANOMALIES

Before detecting OD traffic anomalies, traffic behavior needs to be firstly characterized. To this aim, some OD traffic models need to be fitted, so as to generate predictions against which monitored values can be compared. OD traffic models can be computed for the expected average by predicting two response variables: the *mean* (μ_{od}) and the *standard deviation* (σ_{od}). For the sake of simplicity, hereafter we use just μ and σ in the understanding that those refer to the mean and standard deviation, respectively for a specific OD pair.

Figure 1 illustrates the main steps of the process. Monitoring traffic values for every OD pair are collected from the packet nodes at a given monitoring period, denoted as δ. OD traffic models are fitted with these data (Fig. 1a). Upper and lower bounds, computed as $\mu \pm 3\sigma$, and traffic samples for a given *od* pair and for a typical day are shown in Fig. 1b. Received monitored data can be now compared against its *od* traffic model and those out-of-bound values are considered as *atypical* (Fig. 1c). Notwithstanding, its detection does not entail a traffic anomaly evidence. In fact, the decision of whether an atypical sample is considered as a traffic anomaly cannot be based on just one single sample, but in observing some previous samples and computing a sort of likelihood of that *od* to be anomalous; we call this as *score*. Score values are compared against a defined threshold value (ε_A) and those scores exceeding such threshold are considered as anomalies. An example is depicted in Fig. 1d, where an anomaly is detected after receiving two atypical values and considering some other previous within-bound samples. As pointed out in the introduction, a virtual network reconfiguration can be triggered after an anomaly is detected so as to adapt its topology to the new traffic conditions.

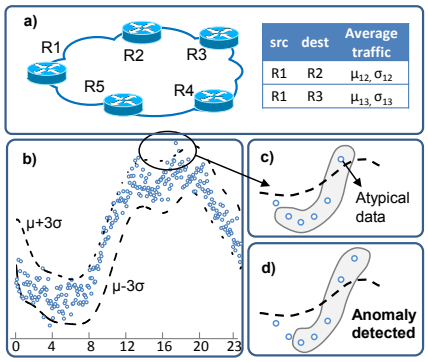


Figure 1: (a) OD models; (b) Samples and boundaries: c) atypical, d) anomaly.

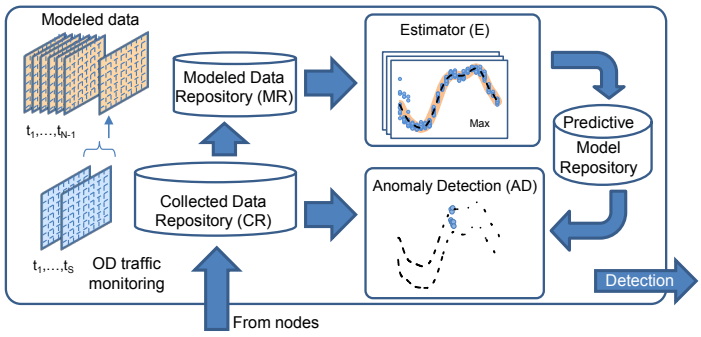


Figure 2. Architecture for OD traffic anomaly detection.

3. OD TRAFFIC ANOMALY DETECTION METHOD

Two different methods for anomaly detection are studied: *traffic-based* and *score-based*, where the methods, already proposed in the literature, are adapted for OD traffic anomaly detection. The adapted *traffic-based* method consists in detecting anomalies after receiving a number of consecutive atypical monitoring data with respect to the $\mu \pm 3\sigma$ confidence interval.

Table 1. Score-based algorithm for traffic anomaly detection.

INPUT: $y(t)$, $\hat{Y}$
OUTPUT: *Anomaly*

```
1:  ŷ(t) ← Normalize (y(t)) (eq. (1) )
2:  remove oldest value from Ŷ
3:  Ŷ(t) ← add (Ŷ, ŷ(t))
4:  if ŷ(t) < 3 (atypical) then
5:      return false
6:  s(t) ← computeScore (Ŷ(t)) (eq. (3))
7:  if s(t) < ε_A then
8:      return false
9:  return true
```

The *score-based* method is a probabilistic classifier with two labels for the response: normal and anomaly. The algorithm (see Table 1) is based on a multi-response model to predict whether a sequence of consecutive traffic data belongs to the normal class or, on the contrary, there is sufficient evidence to declare it as anomalous. Since this method considers previous (not only atypical) traffic monitoring data, it has the capability of potentially anticipating traffic anomalies thus, reducing detection time compared to that of the traffic-based method. The algorithm starts when a traffic monitoring data of a given OD pair $y(t)$ is received in time t; let $\hat{y}(t)$ be the normalized value of $y(t)$ with respect to the average model (equation (1)), where $\mu(t)$ and $\sigma(t)$ are the mean and standard deviation, respectively, of the traffic model for such OD pair at time t.

$$\hat{y}(t) = \frac{y(t) - \mu(t)}{\sigma(t)}, \qquad (1) \qquad \hat{Y}(t) = \left\{\hat{y}(t-i), \forall i \in 0..m-1\right\} \approx N\left(0_{mx1}, I_{mxm}\right), \qquad (2)$$

Note that a normalized value equal to k means that $y(t) = \mu(t)+k\cdot\sigma(t)$. After normalization, $\hat{y}(t)$ is stored in a fixed-size data series $\hat{Y}$ containing the last m normalized traffic data received. Therefore, at a given time t, $\hat{Y}$ contains the following normalized traffic data (eq. (2)), where N represents the multivariate Gaussian distribution with mx1 zero vector mean and $m \times m$ identity covariance matrix. This multivariate distribution is the key result of normalizing traffic by means of μ and σ models. Note that the identity covariance matrix indicates unitary standard deviation for every single $\hat{y}$ value and no correlation between any pair of elements in $\hat{Y}(t)$. Hence, we can conclude that $\hat{Y}(t)$ contains independent and identically distributed random variables each following the univariate standard Gaussian distribution $z \sim N(0,1)$.

According to the aforementioned properties of the normalized traffic, let us define the probability $p(i)=1-P(z \leq |\hat{y}\ (t-i)|)$ as an indicator of how likely is to consider $\hat{y}$ as a normal traffic value. Therefore, we assume that smaller (i.e., less probable) $p(i)$ values will be observed in case of an anomaly. Based on these individual probabilities, we define a score function $s(t)$ to compute how likely is that a data series $\hat{Y}(t)$ does not belong to the normal class. The score $s(t)$ is defined as:

$$s(t) = \frac{1}{\sqrt[m]{\prod_{i=0..m-1}\left[1 - P\left(z \leq \left|\hat{y}(t-i)\right|\right)\right]}} \qquad (3)$$

In view of eq. (3), it is worth noting that the lower the probabilities of $\hat{y}$ variables, the lower the product of probabilities and inversely, the higher the score. To decide whether an anomaly is detected, we simply compare $s(t)$ against the ε_A threshold that normal data series $\hat{Y}(t)$ do not practically exceed. Then, there is sufficient evidence to detect an anomaly in OD pair in time t if $s(t) \geq \varepsilon_A$.

To efficiently implement OD-based traffic anomaly detection methods, we propose the modules depicted in Fig. 2 that are all of them, for the moment, assumed to be placed in the network controller. In such *centralized* architecture, traffic samples are collected from the packet nodes and stored in the collected data repository. Collected data can be conveniently summarized in modeled data, e.g., by computing average values. The Estimator module applies data analytics on samples from the modeled data repository to estimate the specific

models for every OD pair, which are stored in a model repository. Models predict response variables for the average OD traffic (i.e., $\mu(t)$ and $\sigma(t)$). An anomaly detection module is in charge of detecting traffic anomalies; it first verifies whether a just arrived monitored OD traffic value is out of bounds and, only in such case, its current score is computed and compared against threshold ε_A. Upon the detection of an anomalous OD pair, the network controller is notified.

Analyzing the placement of the functions in Fig. 2, it is clear that repositories need to be centralized, since data can be used for several purposes that might require a global view of the network. Regarding model fitting, it can be carried out in the controller from monitoring data collected every 15-minute period. However, that monitoring period would impact on the time to detect OD traffic anomalies and hence, shorter monitoring period would be preferred from that viewpoint. For instance, if the target time to detect traffic anomalies is within the 5 minutes after they appear, the monitoring period cannot exceed that value. In consequence, δ is a key parameter to study since reducing it entails increasing the amount of monitoring data to be conveyed from the network nodes to the collected data repository in the controller.

Aiming at limiting the amount of collected data, in this paper we propose studying the performance of the following monitoring strategies: *i*) the traditional *fixed* monitoring period strategy but reducing its period to accelerate anomaly detection; *ii*) a *dynamic* monitoring strategy, where the monitoring period can be re-programmed during the day; and *iii*) a *reactive* monitoring strategy (*c*:*f*) that uses a coarse monitoring period (*c*) and re-configures it to a finer period (*f*) after detecting the first atypical monitoring data. From the possible combination of methods and strategies, we focus on studying the four most relevant approaches (Table 2): *i*) traffic-based with fixed monitoring (*traffic-fixed*); *ii*) score-based with fixed monitoring (*score-fixed*); *iii*) score-based with dynamic monitoring (*dynamic*); and *iv*) score-based with reactive monitoring (*reactive*).

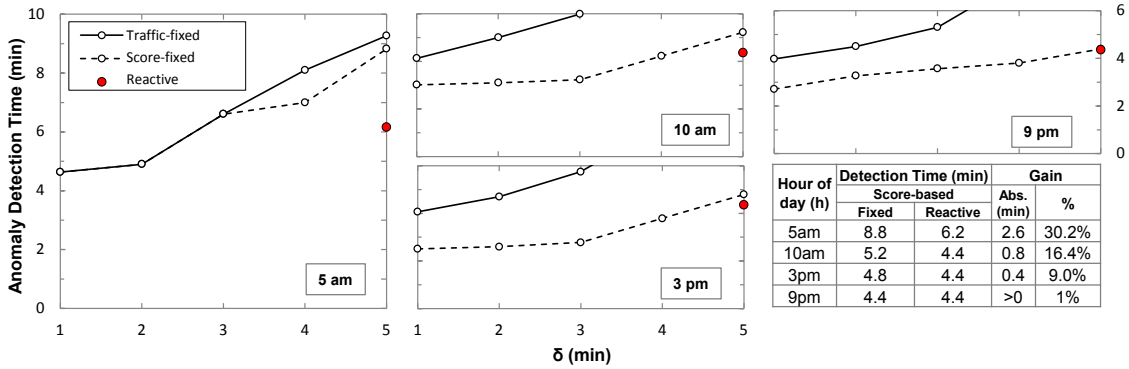
Table 2. Anomaly detection methods and monitoring strategies.

Monitoring Strategy	Detection method	
	Traffic-based	Score-based
Fixed	X	X
Dynamic		X
Reactive		X

4. ILLUSTRATIVE NUMERICAL RESULTS

For evaluation purposes we developed an ad-hoc event-driven simulator in OMNET++ and considered a scenario with ten packet nodes (i.e., 90 OD pairs). OD traffic and traffic anomalies were generated separately and subsequently combined. Traffic is generated as the summation of two different functions: mean and noise, where traffic mean represents a normal day with values varying along day hours and noise is a random function with mean zero and a given standard deviation. Regarding anomalies, they are generated following a pulse function, where the raising front consists of an exponential function and are used as a multiplicative factor over traffic. Anomalies can be configured to be triggered at any specific time and with any specific duration and scaling factor. As an example, an anomaly can be generated to multiply traffic by ×1.5, last for two hours and reach 90% of its maximum value at the first 30 minutes.

Graphs in Fig. 3 plot, for several hours of the day, the anomaly detection time for the considered detection methods and monitoring strategies, where the monitoring period is in the interval [1-5] minutes. We observe that although anomaly detection time varies for the different considered hours of day, detection time increases remarkably with the *traffic-fixed* approach when the monitoring period increases. This is in contrast to the moderated increment achieved by the s*core-fixed* one. In the case of the *reactive* approach, where we assume a (c = 5 : f = 1) min. monitoring strategy, slightly lower detection times with respect to the previous approaches can be observed. The table in Fig. 3 reports the gains in detection time for the studied hours of day, where using a finer monitoring period after an out-of-bound traffic sample is detected provides gains between 1% and 30%.



Hour of day (h)	Detection Time (min)		Gain	
	Score-based		Abs. (min)	%
	Fixed	Reactive		
5am	8.8	6.2	2.6	30.2%
10am	5.2	4.4	0.8	16.4%
3pm	4.8	4.4	0.4	9.0%
9pm	4.4	4.4	>0	1%

Figure 3. Traffic anomaly detection time vs. monitoring period for different hours of a day.

Figure 4 focuses on studying in depth the potentials of the score-fixed and illustrates how anomaly detection depends on different factors such as the changes on traffic volume among the different hours of the day for the same monitoring period. This opens the opportunity to dynamically adapt the monitoring period for different hours of day and achieve the same anomaly detection times (Dynamic monitoring). Note that this is positive

since to achieve low detection times, 1-minute period should be fixed. Hence, by relaxing the monitoring period we are effectively reducing the amount of monitoring data to be collected in the centralized repository.

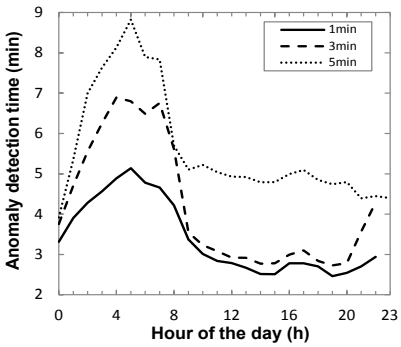


Figure 4. Anomaly detection time vs. h for different δ values.

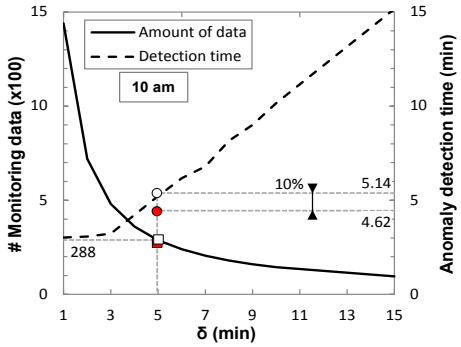


Figure 5. Amount of collected data and anomaly detection time vs. δ.

Figure 5 shows the amount of monitored data to be collected along the day when reducing δ. E.g., assuming a 5 minutes period, 288 monitoring samples per OD and day need to be collected achieving 5.14 and 4.62 min. detection times for the score-fixed and the reactive approaches, respectively.

Even though the different proposed methods for anomaly detection can be improved by changing the monitoring period, the best solution to avoid dealing with huge amount of monitored data is clearly to place the traffic anomaly detection directly into the network node.

5. CONCLUDING DISCUSSION

The above showed that 15-minute monitoring cannot provide the short anomaly detection times required to react against unexpected traffic changes. Consequently, we studied four different approaches mixing detection methods and monitoring strategies and showed that the shortest anomaly detection times are achieved when monitoring every 1 min. Nevertheless, this comes at the cost of collecting and storing large amount of data in management plane.

In view of the above, we propose to bring the proposed data analytics method for OD anomaly detection to the network nodes thus, relaxing data collection from the management plane to the traditional 15-min. period, that can be used for traffic modelling and estimation purposes. Figure 6 and Table 3 detail function placement in the centralized and distributed architectures.

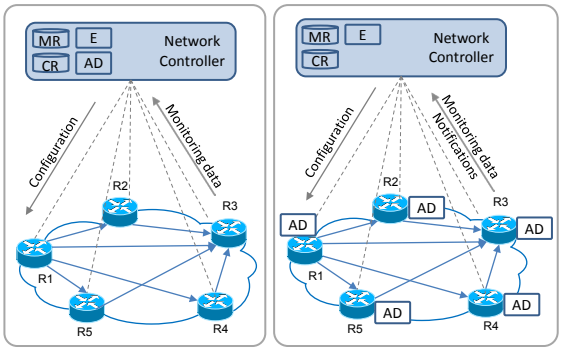


Figure 6. Centralized and distributed architectures.

Table 3. Comparative function placement.

	Centralized	Distributed
Management Plane	• Monitoring freq. program. • Traffic estimation. • OD Anomaly detection.	Traffic estimation.
Node	Monitoring frequency reconfiguration.	OD Anomaly detection.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kum "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 [Invited]," *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 9, pp. A35-A45, 2017.

[3] L. Velasco, A. P. Vela, F. Morales, and M. Ruiz, "Designing, operating and re-optimizing elastic optical networks [Invited Tutorial]," *IEEE/OSA Journal of Lightwave Technology (JLT)*, doi: 10.1109/JLT.2016.2593986, 2017.

[4] C. Liu *et al.*, "OpenMeasure: Adaptive flow measurement and inference with online learning in SDN," in *Proc. IEEE Global Internet Symposium*, 2016.

[5] ITU-T Recommendation M.2120, 2002.

[6] E. Masala, A. Servetti, S. Basso, and J.C. De Martin, "Challenges and issues on collecting and analyzing large volumes of network data measurements," in *New Trends in Databases and Information Systems*, 2014.