Unsupervised real-time detection of BGP anomalies leveraging high-rate and fine-grained telemetry data

Andrian Putina[1], Steven Barth[2], Albert Bifet[1], Drew Pletcher[2], Cristina Precup[2], Patrice Nivaggioli[2] and Dario Rossi[1]
[1]Telecom ParisTech, France – {first.last}@telecom-paristech.fr
[2]Cisco Systems, France – {first.last}@cisco.com

***Abstract*—Recent technology evolution of network equipment allow to continuously stream a wealth of information, pertaining to multiple protocols and layers of the stack, at a very fine spatial-grain and at furthermore high-frequency. Processing this deluge of telemetry data in real-time clearly offers new opportunities for network control and troubleshooting, but also poses serious challenges. In this demonstration, we tackle this challenge by applying streaming machine-learning techniques to the continuous flow of control and data-plane telemetry data, with the purpose of real-time detection of BGP anomalies. In particular, we implement an anomaly detection engine that leverages DenStream, an unsupervised clustering technique, and apply it to telemetry features collected from a large-scale testbed comprising tens of routers traversed by 1 Terabit/s worth of real application traffic.**

I. Introduction

Simple Network Management Protocol (SNMP) has long been the de facto standard to gather relatively coarse information from the network management, control (and coarse data) planes. Consequently, SNMP has been used for anomaly detection for long time [1].

In the SNMP paradigm, the server initiates the data collection from hundreds of devices, with a pull-based approach, at traditionally low frequency (i.e., in the order of minutes). More recently, Model-driven telemetry (MDT) [2] as emerged as an interesting alternative to SNMP: instead of having to periodically poll at a low rate (as in SNMP), under MDT subscribers receive continuous stream of operating state information in a standard structured format (e.g., YANG data models). In addition to supporting periodic export, MDT further enables to trigger data publication when specific conditions are met. While abundance of information is desirable for fine-grained monitoring, however it is necessary to also process MDT data as they are streamed, a challenging task.

In this demonstration, we leverage MDT data for anomaly detection in BGP-only [3] datacenter network. We stress that while anomaly detection is an old field, BGP anomaly detection is not: particularly, the recent survey [4] reveals that machine-learning techniques are only seldom used in BGP anomaly detection. Moreover, whenever machine learning is used (as in [5]–[7]), then without exceptions a *supervised* technique is selected (such as decision trees [5], [7] or support vector machines [6], [7]). Yet, the use of BGP in CSP datacenter is a recent evolution, with a complete different set of dynamics with respect to the more common Internet-case treated in [5]–[7], where additionally the anomaly detection technique is applied to a few tens of features extracted every minute. It is clear that porting any supervised model to a different environment with radically different dynamics, features and timescales is simply meaningless.

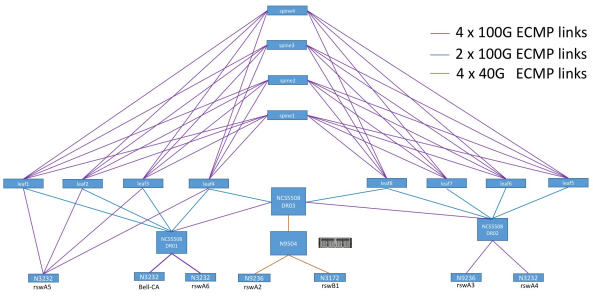

Fig. 1: Testbed topology. Notice that in the clos topology, purple links each carry 4× 100Gbps fibers.

In this demonstration, we leverage *unsupervised* techniques (such as DenStream [8]) to process MDT data in real-time for online outlier detection. We make our code available as open-source software [9] – which is interesting per se as, to the best of our knowledge, there is no fully functional DenStream implementation available. Additionally, we plan to make the datasets generated for the demonstration available as well – which has independent value, since our large-scale testbed represent state-of-the art [3] datacenter networks with hundreds of servers interconnected by tens of devices, traversed by a 1 Tbps aggregated traffic.

II. Demonstration at a glance

A. Testbed and dataset

We have setup a full-scale testbed comprising comprises 23 nodes in Cisco premises that will be accessed remotely during the demo. The testbed, shown in Fig.1, replicates a traditional clos topology of a Content Service Provider (CSP) datacenter. On the physical level, it comprises 8 leaf nodes interconnected via 4 spine nodes. For redundancy, each leaf is connected to each spine via 4× 100Gbps fiber links. On the operational level, the datacenter is designed with BGP as the only routing protocol, following guidelines in [3].

Real application mixtures are generated from servers in the racks (not shown in the picture) connected to the ToR switches

(the Nexus 2/3/5000 and 9000 series) to generate up to 1 Tbps of aggregated traffic. Therefore, whereas the testbed does not involve real users, it do however uses real equipment, protocols and applications used in production networks.

Each of the 23 testbed nodes generates about 1,000 MDT control and data plane features every second. During the course of the demonstration, different types of BGP anomalies (e.g., BGP flapping, leaks, table clears, etc.) are injected at different nodes. Controlling the anomaly allows to construct a reliable *ground truth (GT)*, which is not used to train off-line supervised machine learning algorithms, but merely for the purpose of assessing the main performance indicators of *online unsupervised techniques*.

Given the scale of the testbed, and the magnitude of the traffic, we believe that there is value for not only *processing* telemetry data we generate during the demo in real-time, but also *storing* it a dataset for future use. We thus plan to release MDT data as open-source *dataset* to the community: annotated with GT information, these MDT datasets which will help to replicate our experiments, as well as propose and contrasts alternative anomaly detection techniques to ours.

B. Anomaly detection engine

Given that supervised models can be hardly generalized, we advocate the use of *unsupervised* techniques, where the model is constructed over the data without previous knowledge, and continuously updated at each new data sample. The scientific literature abunds with simple (K-Means) to more sophisticated (DBScan) clustering techniques. A lesser known approach is DenStream [8], proposed by the same authors of DBscan, that is more apt to our case as it applies to an evolving data stream: the algorithm uses a damped window model to weight the samples via an exponential *fading function* $f(t) = 2^{-}\lambda t$. DenStream allows to discover clusters of arbitrary shape by maintaining a set of normal and outlier micro-clusters: these micro-clusters are maintained incrementally applying at each time stamp the fading factor to the weighted linear sum (WLS) and weighted squared sum (WSS) of the interior points. Anytime a new datapoint is received, DenStream either merges it to a normal/outlier micro-clusters, or creates a new outlier micro-cluster, triggering an alarm.

As in our case there are features that may react with a delay (e.g., the data traffic features will take some time to reflect a BGP path change) we batch together subsequent alarms, raising an anomalous events only when DenStream raises K consecutive alarms: in our design, K is a configuration parameter trading off timeliness vs false alarm rate. Given that our DenStream implementation has value on its own (i.e., a DenStream implementation is not available as open-source software to the best of our knowledge) we make it available on GitHub [9].

C. Visualization

We develop a real-time Javascript based visualization engine that collects, processes and visualize results of the anomaly detection engine, of which a snapshot is reported in Fig. 1. In particular, the visualization depicts:

- the engine precision and recall (top-bar: current selected order K; top-right plot: all orders $K \in [1, 5]$)
- statistics about the DenStream model (top-right: the number of core and outlier micro clusters)
- allowing to interactively explore the data by selecting pairs of features among the available ones (bottom-left: scatter plot of selected features; bottom-right: time evolution of selected features).

Additionally, the visualization engine allows user to inject controlled anomalies in real-time, and to tweak fundamental DenStream parameters (such the exponent λ, the micro-cluster threshold ϵ) to gauge the system response in an intuitive way.

Fig. 2: Snapshot of the actual dashboard. A preliminary video of the dashboard in action is available at [10] .

ACKNOWLEDGMENTS

This work has been carried out at LINCS (http://www.lincs.fr) and benefited from support of NewNet@Paris, Cisco's Chair "NETWORKS FOR THE FUTURE" at Telecom ParisTech (http://newnet.telecom-paristech.fr).

REFERENCES

[1] M. Thottan and C. Ji, "Anomaly detection in ip networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2191–2204, 2003.

[2] I. F. for T. (2017) Telemetry. http://www.telemetry.org/. [Online]. Available: http://www.telemetry.org/

[3] P. Lapukhov, A. Premji, and J. Mitchell, "Use of bgp for routing in large-scale data centers," RFC7938, Tech. Rep., 2016.

[4] B. Al-Musawi, P. Branch, and G. Armitage, "BGP Anomaly Detection Techniques: A Survey," in *IEEE Commun. Surveys & Tutorials*, 2016.

[5] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal, "An internet routing forensics framework for discovering rules of abnormal bgp events," *ACM SIGCOMM Comp. Comm. Rev.*, vol. 35, no. 5, pp. 55–66, 2005.

[6] I. O. de Urbina Cazenave, E. Köşlük, and M. C. Ganiz, "An anomaly detection framework for bgp," in *IEEE INISTA*, 2011, pp. 107–111.

[7] N. M. Al-Rousan and L. Trajković, "Machine learning models for classification of bgp anomalies," in *HPSR,*. IEEE, 2012, pp. 103–108.

[8] M. Ester, F. Cao, and A. Z. Weining Qian and, "Density-based clustering over an evolving data stream with noise," *SIAM Conference on Data Mining*, vol. 1, pp. 322–337, 2006.

[9] https://github.com/anrputina/OutlierDenStream.

[10] http://newnet.telecom-paristech.fr/index.php/telemetry.