ENDEAVOUR: D3.1: Monitoring requirements

Ignacio Castro, Eder Leao Fernandes, Gianni Antichi, Mitch Gusat, G. Kathareios, Steve Uhlig, Philippe Owezarski

ENDEAVOUR: Towards a flexible software-defined network ecosystem

ENDEAVOUR

Project name	ENDEAVOUR
Project ID	H2020-ICT-2014-1 Project No. 644960
Working Package Number	3
Deliverable Number	3.1
Document title	Monitoring Requirements
Document version	0.2
Author	Castro, Fernandes, Antichi, Gusat, Kathareios, Uhlig, Owezarski
Date	3 July 2015
Status	*Public*

Revision History

Date	Version	Description	Author
3 July	0.1	First draft	Castro, Fernandes, Antichi, Gusat, Kathareios
3 July	0.2	Minor changes	Uhlig
9 July	1.0	Add contributions on monitoring and security	Owezarski

Contents

Executive Summary

The declared mission of ENDEAVOUR is to advance the Internet interconnection model to a new paradigm through the introduction of Software Defined Networking (SDN) technology at one of the central elements of the Internet architecture, the Internet Exchange Point (IXP). While SDN enables a whole new set of services, the implementation of these novel capabilities has additional monitoring requirements. This deliverable surveys the monitoring needs relevant for the new capabilities that SDN brings to the IXP. While the high dynamism that SDN brings requires novel needs in terms of monitoring, it also enables new monitoring capabilities through a more extensive and flexible data gathering. To better understand the opportunities and challenges that SDN-enabled monitoring introduces, this deliverable does a per use-case analysis of monitoring requirements. While deliverable D.4.1 discusses these use cases in detail, the focus here is in identifying the monitoring needs, the available methods, and their limitations. To examine the challenges introduced by the monitoring requirements, we carry out an analysis of the state of the art in the monitoring techniques related to our goal of an SDN enabled IXP. In doing so, we first propose a novel taxonomy to classify existing techniques, and then survey the main techniques employed by networks in general, SDNs, clouds, and how monitoring is also used for security purposes.

1 Introduction

The deployment of SDN at IXPs leads to new monitoring requirements, to meet the potentially stringent requirements posed by the Service Level Agreements (SLA) offered. As the network speed increases, the monitoring capabilities need to cope with the corresponding growing traffic volumes. In addition to that, SDN comes with new challenging elements: the dynamic nature of SDN, driven by constant automation and broader network's intelligence, requires the evolution of the current monitoring framework. Moreover, monitoring must be integrated with the SDN control plane. The algorithms controlling the network, implemented as applications within the SDN stack, will rely on real-time data collected by monitoring instances.

General requirements include monitoring of real-time changes, as well as an architecture that scales with the network capacity. As SDN comes to dynamically change the configurations of today's networks, the current practice of scheduled monitoring verification is insufficient. Furthermore, the

time needed to verify configuration files and the data plane state are likely to be incompatible with the dynamism inherent to SDN-based operations.

To better understand monitoring requirements and its challenges, we analyse these two issues separately. First we examine the monitoring requirements for the different use cases of SDN-enabled IXPs as proposed in the literature. While Deliverable 4.1 will make an in depth study of such cases, the analysis here is limited to the type of information that needs to be gathered to adequately monitor the correct implementation of the policies chosen by the ASes. Then, we explore the challenges resulting from these novel requirements. In doing so, we first provide a classification of the state of the art in monitoring techniques. We then discuss how these techniques are used to cope with the challenges of monitoring highly dynamic systems. Finally, we study how monitoring techniques have been leveraged for security purposes.

2 Monitoring requirements

2.1 Load Balancing

Internet content providers typically load balance their clients requests across clusters of servers by manipulating the Domain Name System (DNS). This approach is cost efficient, because it does not requires specialized middle boxes. It comes however at the cost of a slow response to failures due to DNS caching [50]. Solutions to this problem are rather limited, e.g., reducing the DNS Time To Leave (TTL) values leads to more frequent DNS cache misses and therefore higher latency to obtain the DNS responses.

SDN programmability can be leveraged at the IXP [27] to overcome the limitations of content-aware traffic engineering based on DNS tweaking [24, 42]. While load balancing can be handled by taking advantage of SDN flexibility and programmability, legacy layer 2 IXPs typically resort to costly and more complex network equipment and protocols, such as Label Path Switching (LSP) and Virtual Private LAN Services (VPLS).

Monitoring the fabric data plane is fundamental to ensure that the chosen load balancing policy is adequately implemented [27]. In addition, the results obtained from the measurements can also be used to trigger different solutions as new requests arrive. In this context, it is necessary to monitor the volume of traffic per IP destination sent out to a given physical port.

2.2 Inbound and Outbound Traffic Engineering

BGP destination-based routing constrains how IXP members control the inbound traffic in their networks. Although IXP customers might take advantage of some BGP attributes to influence how packets enter their Autonomous Systems(ASes) [44, 12, 45], the performance of these techniques is very limited [27]. Due to these BGP restrictions, Gupta et al. presents an SDN solution for customers who exchange packets at an IXP and want to have a better control on their incoming traffic. Using SDN-enabled switches, e.g., OpenFlow switches, inbound traffic can be controlled using flow forwarding rules based on packets source IP address or input port.

Because outbound traffic engineering does not involve the alteration of route announcements (outside the local AS) to influence how other ASes reach a given destination, control over the egress traffic with BGP is much easier than inbound traffic engineering. By identifying specific routes and tweaking their local preference, an AS can change its default forwarding policy. Nevertheless, it is still limited by the destination-based nature of BGP routing [56]. On the other hand, in an SDN scenario an IXP member could perform outbound traffic engineering based on a specific application through the matching of specific layer four ports.

Outbound and inbound traffic engineering are highly similar from a monitoring requirements point of view. To ensure the correct implementation of policies, the system needs to monitor the amount of packets per physical port (i.e., IP source/destination address pairs, layer four ports, etc.).

2.3 Peering

By introducing multiple approaches that go well beyond the nowadays exclusive BGP-based routing mechanism, SDNs greater flexibility brings peering at the IXP to a new dimension. For instance, routing based on the packet's layer 4 ports, allows finer grained decisions on the peering policies as it enables ASes to peer for specific types of applications, such as video streaming. By enlarging the scope of peering to new relationships based on specific packet fields, an SDN-enabled IXP can create richer relationships and business cases. This new range of capabilities may result in more complex policies, imposing substantial information needs both for the networks who benefit from it as well as for the IXP who supports them.

As a previous step to filtering the traffic by other packet fields rather than the destination IP address, it is first necessary to ensure that the AS with which the peering is established is the right one. This extent is done by

the peering ASes, which additionally configure the more fine grained policies in the SDN-enabled IXP. Since the ASes are in charge of the control plane aspects, the only monitoring requirements are at the data plane level [27]. Ensuring the proper operation of configured policies, requires monitoring whether the packets appropriately match the right fields.

2.4 Overlay monitoring

Overlay Virtual Networks (OVNs) provide many benefits to the underlying network, such as better load balancing, simplicity and resiliency. However, since multiple encapsulation layers can be in use at the same time, OVNs hinder the effectiveness of the monitoring process. In the context of the IXP, where the translation between different tunnels will ideally take place (see D.4.1), the monitoring process must be efficiently performed regardless of the OVN used.

For overlays where the control plane is used for MAC learning (as is the case of MP-BGP for EVPN-enabled VXLAN), the monitoring process could keep track of the exchange of routing entries. The information gathered could potentially be fed to the ASes and provide them with valuable information to make future control plane decisions. Some desired measurements would be the amount of traffic per OVN subnet (VNI in VXLAN) and/or a traffic matrix (see 3.2.1) among tunnel endpoints.

The basic requirement for overlay monitoring on the data plane is the ability to implement Deep Packet Inspection (DPI), allowing the discovery of all encapsulation layers, so as to allow traffic classification and measurements per overlay network. This extent can only be realized within the constraints resulting from encryption, which result in large computational requirements and significant privacy concerns.

2.5 Security

Monitoring the network status is the first step to prevent attacks. When a network detects a security threat it can react by filtering out the unwanted traffic, i.e., passing or dropping the traffic according to a previously decided criteria. For instance, blackholing was recently introduced and implemented at various IXPs [1]. IXPs employ blackholing to discard unwanted traffic, for example during a Distributed Denial of Service (DDoS) attack. Another key example is the possibility to prevent the Address Resolution Protocol (ARP)

[1] https://www.de-cix.net/products-services/de-cix-frankfurt/blackholing/ [Last accessed 22.06.2015]

storm effect. This can be done by filtering out location discovery traffic at the exchange when the amount of ARP packets rapidly increases (because of network failure or network attacks). A common practice to reduce the amount of location discovery traffic in today's IXPs is the use of the ARP sponge server [1]. By installing filtering policies directly in the OpenFlow-enabled switches, SDN provides an alternative. Excessive amount of ARP requests could also be handled by the controller [39]. Another possibility is to have the controller directly answering ARP requests or completely avoid broadcast through direct forwarding to the IP destination of the ARP request [9].

The monitoring requirements strictly depend on the security aspects that the IXP wants to tackle and its dimension. As an example, ARP storm effect prevention at large IXPs is a necessary feature. In this case, the SDN controller should be able to monitor the amount of location discovery traffic into the network to trigger appropriate filtering policies when it exceeds a predefined threshold.

Another case is the detection of DDoS attacks, a task for which Open Flow-enabled switches are particularly useful. Because a DDoS attack generally attempts to interrupt or suspend services of a host connected to the Internet by overwhelming its ability to handle the requests, constant monitoring of the traffic towards a given target (e.g., a layer 3 address) is the first step to detect and filter out these kind of attacks. Going further, it is likely required these detection and filtering mechanisms to leverage on cognitive systems able to autonomously analyse the traffic in real time, for learning on new traffic and attack classes. Such solutions often rely on machine learning techniques [38] [8]. Such detection then requires small amount of human resources and can be effective in real-time.

The DDoS detection solutions currently available generally focus on a single link traffic, whereas it is required to enforce the same policy on the full network. Thanks to its function virtualization capabilities, the SDN concept can allow the distributed detection process to be centralized, for a maximal efficiency.

2.6 Enabling Services

2.6.1 Traffic Steering

Traffic steering refers to the redirectioning of traffic towards middleboxes within a network based in some predefined rule.

Middleboxes are commonly placed in strategic points of a network to

provide security, monitoring, and other, services. Because of the prohibitive cost of placing middleboxes ubiquitously, these ASes manipulate traffic to make it pass through the desired middleboxes. One example is the announcement of BGP prefixes to direct packets to a network appliance where the traffic will be analyzed. This mechanism often gets more traffic than necessary and is also error prone, since a misconfiguration could redirect the wrong packets to the middleboxes. An approach to address these issues is SDN. With SDN-enabled switches, redirectioning of flows subsets is simpler and can also be based on specific fields besides the IP address destination.

The monitoring requirements for traffic steering strictly depend on the technique enabled. In most of the cases, the network administrator decides to redirect a given flow to a middlebox when a specific event in the network occurs (i.e., a new flow is seen in the network). Whenever this event is control plane-related (i.e., new BGP announcements), monitoring at the control plane level is needed.

2.6.2 Routing as a Service (RAS)

Routing as a Service (RAS) was a SDN predecessor, which already presented a clear separation of the control and data planes [34]. In RAS, the task of computing the route between source and destination is outsourced to an external entity. The advent of SDN, has revitalized the idea of RAS as a powerful tool to change the routing picture of the Internet [37]. As peering fabrics, IXPs seems to be natural aspirants to embrace new routing mechanisms. Currently, IXP members peer among them through BGP sessions originated from their own routers. By supplying routing services, IXPs could free their members from the drawbacks of BGP and push a new era of innovation on Interdomain routing.

Because of the clear decoupling on the tasks of route calculation and packet forwarding, monitoring requirements for RAS involves both control and data planes. Control plane monitoring in RAS involves two basic aspects:

- **Forwarding Information Base (FIB).** The routes computed in the control plane are translated to forwarding information into the data plane. Thus, it is important to monitor the control plane FIB in order to verify whether the SDN-enabled switches reflect the correct routes.
- **Convergence time.** The convergence time of the route calculation algorithm is an important metric to understand the overhead of outsourcing routing. This information gives useful feedback to calibrate

the configuration of hello-based protocols, like OSPF, or even to switch to more efficient route calculation mechanisms.

Data plane monitoring requirements imply the two following elements:

- **FIB.** To ensure consistence with the routes calculated by the control plane, the forwarding information in the switches need to be monitored.
- **Topology changes.** Depending of the route calculation algorithm, topology changes in the data plane may affect the computation on the control plane. For this reason the topology must be monitored in order *to allow the control plane* to promptly react to modifications. Also, network operators could benefit from the history of topology changes to identify possible network bottlenecks and typical points of failure.

3 Challenges in monitoring

While the previous section examined the type of information that must be collected to monitor and ensure a correct implementation of the different capabilities enabled by SDN, this section analyzes the challenges to implement such monitoring. To better understand the state of the art in monitoring techniques, we first propose a taxonomy for its classification. Then we elaborate on the current and proposed monitoring techniques at different levels: for networks in general, for SDNs, and finally for cloud computing. Note that cloud monitoring is included here not only due to its close relationship with SDNs, but also because its great relevance at the IXPs. Finally we also discuss how monitoring is currently leveraged for security purposes.

3.1 Taxonomy of monitoring methods

With a large body of existing knowledge in monitoring, we first provide a practical classification of the state of the art. This taxonomy takes into account the dual perspective of SDN and IXPs, and provides a classification along functional, topological, and methodological dimensions.

Functional: The monitoring system of a network is aimed at one or more of the following functions:

1. **Performance and QoS-compliance/SLA enforcement:** metric collection-based monitoring aims at collecting volume measurements

and statistics for overall throughput, local and/or global delay. This helps to: 1) quantify the performance of, e.g., newly instantiated Virtual Machine (VM) deployment to produce an initial benchmark that can determine whether the deployment meets the acceptable performance; or/and 2) examine the performance of a certain deployment to determine if/how often the performance drops under the acceptable performance requirement.

2. **Management:** this class of monitoring aims at the definition, enforcement and reporting of access control lists for SDN/Overlay encapsulations (tunnels, VETPs) and/or input for load balancers (ECMP/LAG, BGP, DNS).

3. **Security:** this class of monitoring assists in attack, intrusion and DoS detection and firewalls. Monitoring mechanisms perform traffic characterization and automatic behaviour classification to identify malicious traffic and prevent attacks.

Topological: According to the scope of the monitoring process in a network, it can be classified as:

1. **Local**: typically performed at the core of the network, at each graph vertex, it measures queue occupancy (port, link or interface).

2. **Path**: performed at the edge of the network, it measures throughput and latency on one or more graph paths.

3. **Global**: more recent methods aim at global monitoring, measuring all the components of the network graph, creating congestion matrices, heatmaps [4].

Methodological: Based on the methodology used for measuring or estimating the appropriate metrics, monitoring can be classified as:

1. **Sampling**: typically performed at one graph vertex (queue, port, link etc.), it can be direct (measuring absolute values of various metrics) or indirect (measuring the delta between 2 measurements)

2. **Packet capture (Pcap)**: performed at one core vertex or edge by capturing traversing packets (incoming and outgoing).

3. **Probing/Telemetry**: Performed edge-to-edge, aims at collecting statistics for the interconnecting path (RTT, drop rate etc.).

4. **Statistical analysis**: typically performed offline (inferential, tomography, logs post-processing, etc.).

3.2 Existing monitoring techniques

3.2.1 General Network Monitoring

Despite the extensive literature on network monitoring, telemetry and topography, the current state of the art in hardware network monitoring has remained limited to sampling a few, possibly isolated, links with a granularity in the 0.01s to 1s range such as sFlow [40], NetFlow [15] and SNMP [11].

While traffic matrix-based approaches facilitate decision making based on information gathered from the monitoring and measurements within the network in sFlow [59], IPFIX [14], or Netflow [16, 21] most of them suffer from limited visibility of port based counters.

3.2.2 SDN Monitoring

SDN enables novel monitoring capabilities but also imposes new monitoring requirements. While Open Flow and OpenStack offer new monitoring capabilities and APIs (e.g., richer per flow state, new counters and statistics, etc.) [41], SDN introduces new elements (overlays, tunnels, hypervisors and container/dockers, vswitches and vNICs) whose monitoring proves to be challenging [5, 18].

While traffic matrices are crucial for capacity planning, traffic engineering and routing protocol configuration [57, 54], traffic matrix estimation is problematic [64]. To address the limitations of current traffic matrix estimation methods in SDNs, new proposals such as [63, 19, 58, 26] have emerged.

3.2.3 Cloud monitoring

Cloud monitoring is crucial [60] to accurately quantify the performance provided by the infrastructure. While new monitoring techniques are being continuously proposed [32, 7, 31, 30, 4], these techniques typically face significant challenges with regard to scale, rapidity, detection, localization, and diagnose of performance problems [23].

3.2.4 Overlay monitoring

The popularity of tunnels and overlays further complicates the challenges of SDN monitoring at the IXP. High volumes of encapsulated traffic introduces further complexity by requiring DPI techniques to enable monitoring.

The orchestration, management, load balancing, protection and isolation of the virtualized systems of today's cloud depend on the timely access to datacenter's internal state (e.g., load, occupancy, utilization), including all the layers of the physical and virtual components [60, 61, 4].

With an overwhelming variety of virtualization techniques [13, 35, 52, 25, 10, 36], VXLAN is emerging as the de-facto standard for the future of SDN-based OVN/tunneling [2] for datacenter networks.

While scalability remains limited, an emerging IETF standard [29, 48] that uses MP-BGP for MAC learning in the control plane addresses the problem and extends the VXLAN across a WAN [2]. This solutions are particularly relevant as they open the door to the creation of multi-cloud services and platform-neutral "super"-overlays at SDN-enabled IXPs.

Cloud Transports and Tunnels Optimization Although the Internet is currently dominated by TCP/UDP, virtualized datacenters move towards performance-optimized transport protocols. With TCP suffering from excessive limitations for the highly demanding virtualized datacenters, new protocols have been proposed. In monitoring congestion Datacenter TCP (DCTCP) [3], a TCP transport protocol developed by Microsoft for datacenter networks, goes beyond TCP capabilities by reacting to the extent of congestion and not just to its mere presence. This finer level of control allows DCTCP to operate with very low buffer occupancies while simultaneously achieving high throughput. Multi-path TCP (MPTCP) [46, 49] is another modification of TCP. MPTCP can outperform TCP [46] by offering path redundancy through the simultaneous use of several IP-addresses/interfaces. The Fast and Secure Protocol (FASP) [22] overcomes the performance bottleneck of TCP when moving massive data, particularly for WANs with large bandwidth, high round-trip time and packet loss. zFabric [17], an SDN-based tunneling transport mechanism built on zOVN [18], combines the ubiquity of TCP with the performance of UDP and RDMA, resulting in order of magnitude lighter protocol stacks. This advances raise the question of whether "losslessness" can be extended beyond a single datacenter/PoD to an SDN-enabled IXP.

3.3 Security via monitoring

Network traffic monitoring has become an essential means for detection of network attacks in today's Internet. The principal challenge in detecting network attacks is that these are a moving target. It is not possible to know the different attacks that an attacker may launch, because new attacks as

well as new variants of already known attacks are continuously emerging. Indeed, attacks have become both increasingly numerous and sophisticated over the years [28]

Leveraging on traffic monitoring, two different approaches are by far dominant in current research community and commercial detection systems: signature-based detection and anomaly detection. Despite being opposite in nature, both approaches share a common downside: they rely on the knowledge provided by an expert system, usually a human expert, to do the job. We shall therefore refer to them as knowledge-based detection approaches.

On the one hand, signature-based detection systems [47] are based on a extensive knowledge of the particular characteristics of each attack, referred to as its "signature". Such systems are highly effective to detect those well-known attacks which they are programmed to alert on. However, they cannot defend the network against new attacks, simply because they cannot recognize what they do not know. In addition, building new signatures involves manual inspection by human experts, which is not only very expensive and prone to errors, but also introduces an important latency between the discovery of a new attack and the construction of its signature. In a network scenario where new attacks are constantly appearing, such a manual process imposes a serious bottleneck on the defense capabilities of the network.

On the other hand, anomaly detection [53, 6, 51] relies on the existence of normal-operation traffic instances to build a baseline-profile, detecting anomalies as traffic activities that deviate from it. Such an approach permits to detect new kinds of network attacks not seen before, because these will naturally deviate from the constructed baseline. Nevertheless, anomaly detection requires training to construct normal-operation profiles, which is time-consuming and depends on the availability of purely anomaly-free traffic datasets. Labeling traffic as anomaly-free is expensive and hard to achieve in the practice, since it is difficult to guarantee that no anomalies are hidden inside the collected traffic. Additionally, it is not easy to maintain an accurate and up-to-date normal-operation profile, particularly in a dynamic and evolving context where new services and applications are constantly emerging.

Motivated by the limitations of knowledge-based approaches, a new research area has emerged in the last years, based on a diametrically opposite philosophy for detection of anomalous traffic events: Unsupervised Anomaly Detection. Instead of relying on a previously acquired knowledge on the characteristics of network attacks or on the baseline-traffic behav-

ior, unsupervised detection uses data-mining techniques to extract patterns and uncover similar structures "hidden" in unlabeled traffic of unknown nature (attack or normal-operation traffic). Some methods for unsupervised detection of network attacks have been proposed in the past [43, 20, 33]. Practically, attack and intrusion detection, DoS detection, Firewall (h/w or as NFV) [62] mechanisms have been proposed for traffic characterization and automatic behaviour classification (to identify malicious traffic). New traffic analysis techniques offer higher capabilities which lead to better intrusion detection [55] while still providing a lightweight approach [61].

4 Conclusions

By introducing SDN technology at large IXPs, ENDEAVOUR strives to shift the Internet interconnection model to a new, more advanced paradigm. The whole new range of capabilities enabled by SDN technologies is accompanied by new monitoring requirements as well as monitoring possibilities. This deliverable surveys the monitoring needs, opportunities, and challenges that SDN brings to the IXP. We began by studying which are the monitoring requirements for the use cases analysed in Deliverable 4.1. Then, to understand the new opportunities and challenges, this deliverable first examined what are the novel requirements to implement the promised new capabilities enabled by SDN, and then explored the challenges by reviewing the state of the art in monitoring techniques.

References

[1] AMS-IX controlling arp traffic on ams-ix platform. `https://ams-ix.net/technical/specifications-descriptions/controlling-arp-traffic-on-ams-ix-platform`. Accessed: 2015-06-24.

[2] Learn About VXLAN in Virtualized Data Center Networks. http://www.juniper.net/techpubs/en_US/learn-about/LA_VXLANinDCs.pdf, 2015.

[3] Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitu Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. DCTCP: Efficient Packet Transport for the Commoditized Data Center. In *Proc. SIGCOMM*, New Delhi, India, Aug 2010.

[4] Andreea Anghel, Robert Birke, and Mitch Gusat. Scalable high resolution traffic heatmaps: Coherent queue visualization for datacenters. In *Traffic Monitoring and Analysis*, pages 26–37. Springer, 2014.

[5] Katherine Barabash, Rami Cohen, David Hadas, Vinit Jain, Renato Recio, and Benny Rochwerger. A Case for Overlays in DCN Virtualization. In *Proc DCCAVES*, San Francisco, CA, Sep 2011.

[6] P. Barford, J. Kline, D. Plonka, and A. Ron. A Signal Analysis of Network Traffic Anomalies. In *International Measurement Workshop*. ACM, 2002.

[7] SM Batraneanu, A Al-Shabibi, MD Ciobotaru, M Ivanovici, L Leahu, B Martin, and SN Stancu. Operational model of the atlas tdaq network. *IEEE Trans. Nuclear Science*, 55(2):687–694, 2008.

[8] M. Bhuyan, D. Battacharyya, and J. Kalita. Network anomaly detection: methods, systems and tools. *IEEE communications surveys and tutorials*, 16(1), 2014.

[9] V. Boteanu. Minimizing ARP traffic in the AMS-IX switching platform using OpenFlow, 2013.

[10] Martín Casado, Teemu Koponen, Rajiv Ramanathan, and Scott Shenker. Virtualizing the network forwarding plane. In *Proc. ACM PRESTO*, page 8. ACM, 2010.

[11] J. D. Case, M. Fedor, M. L. Schoffstall, and J. Davin. Simple network management protocol (snmp), 1990.

[12] Rocky KC Chang and Michael Lo. Inbound Traffic Engineering for Multihomed ASs using AS Path Prepending. *IEEE Network Magazine*, 19(2):18–25, 2005.

[13] NM Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862–876, 2010.

[14] B. Claise. Specification of the ip flow information export (ipfix) protocol for the exchange of ip traffic flow information. 2008.

[15] Benoit Claise. Cisco systems netflow services export version 9. *Internet Engineering Task Force, Internet Draft*, 2004.

[16] Benoit Claise. Cisco systems netflow services export version 9. 2004.

[17] Daniel Crisan, Robert Birke, Nikolaos Chrysos, Cyriel Minkenberg, and Mitch Gusat. zfabric: How to virtualize lossless ethernet? In *Proc. IEEE CLUSTER*, pages 75–83. IEEE, 2014.

[18] Daniel Crisan, Robert Birke, Gilles Cressier, Cyriel Minkenberg, and Mitch Gusat. Got Loss? Get zOVN! In *Proc. SIGCOMM*, Hong Kong, China, Aug 2013.

[19] A.R. Curtis, Wonho Kim, and P. Yalagandula. Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection. In *Proc. INFOCOM*, pages 1629–1637, Apr 2011.

[20] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo. A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data. *Applications of Data Mining in Computer Security, Kluwer Publisher*, 2002.

[21] Cristian Estan, Ken Keys, David Moore, and George Varghese. Building a better netflow. In *Proc. SIGCOMM*, SIGCOMM '04, pages 245–256. ACM, 2004.

[22] Xingzhe Fan and Michelle Munson. Petabytes in motion: Ultra high speed transport of media files a theoretical study and its engineering practice of aspera fasp over 10gbps wans with leading storage systems. In *SMPTE Conferences*, volume 2010, pages 2–13. Society of Motion Picture and Television Engineers, 2010.

[23] Danyel Fisher, David Maltz, Albert Greenberg, Xiaoyu Wang, Heather Warncke, George Robertson, Mary Czerwinski, et al. Using visualization to support network and application management in a data center. In *Proc. INM*, pages 1–6. IEEE, 2008.

[24] B. Frank, I. Poese, G. Smaragdakis, S. Uhlig, and A. Feldmann. Enabling content-aware traffic engineering. *ACM CCR*, 42(4):21–28, 2012.

[25] J Gross, T Sridhar, P Garg, C Wright, and I Ganga. Geneve: Generic network virtualization encapsulation. *Internet Engineering Task Force, Internet Draft*, 2014.

[26] Naman Grover, Nitin Agarwal, and Kotaro Kataoka. liteflow: Lightweight and distributed flow monitoring platform for sdn. In *Proc. IEEE NetSoft*, pages 1–9, Apr 2015.

[27] Arpit Gupta, Laurent Vanbever, Muhammad Shahbaz, Sean P Donovan, Brandon Schlinker, Nick Feamster, Jennifer Rexford, Scott Shenker, Russ Clark, and Ethan Katz-Bassett. SDX: A Software Defined internet Exchange. In *Proc. SIGCOMM*, pages 551–562. ACM, 2014.

[28] S. Hansman and R. Hunt. A Taxonomy of Network and Computer Attacks. *Computers and Security*, 24(1), 2005.

[29] Aldrin Isaac, Nabil Bitar, Jim Uttaro, Rahul Aggarwal, and Ali Sajassi. Bgp mpls-based ethernet vpn. Technical report, 2015.

[30] Srikanth Kandula, Ratul Mahajan, Patrick Verkaik, Sharad Agarwal, Jitendra Padhye, and Paramvir Bahl. Detailed diagnosis in enterprise networks. *ACM CCR*, 39(4):243–254, 2009.

[31] Srikanth Kandula, Sudipta Sengupta, Albert Greenberg, Parveen Patel, and Ronnie Chaiken. The nature of data center traffic: Measurements & analysis. In *Proc. IMC*, IMC '09, pages 202–208, New York, NY, USA, 2009. ACM.

[32] Ramana Rao Kompella, Kirill Levchenko, Alex C Snoeren, and George Varghese. Every microsecond counts: tracking fine-grain latencies with a lossy difference aggregator. In *ACM CCR*, volume 39, pages 255–266. ACM, 2009.

[33] A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies Using Traffic Feature Distributions. In *SIGCOMM*. ACM, 2005.

[34] Karthik Lakshminarayanan, Ion Stoica, and Scott Shenker. Routing as a service. Technical Report UCB/CSD-04-1327, EECS Department, University of California, Berkeley, 2004.

[35] M. Mahalingam, D. Dutt, K. Duda, P. Agarwal, L. Kreeger, T. Sridhar, M. Bursell, and C. Wright. VXLAN: A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks. Internet draft, Internet Engineering Task Force, Aug 2011.

[36] Jayaram Mudigonda, Praveen Yalagandula, Jeffrey C. Mogul, Bryan Stiekes, and Yanick Pouffary. NetLord: A Scalable Multi-Tenant Network Architecture for Virtualized Datacenters. In *Proc. SIGCOMM*, Toronto, Canada, Aug 2011.

[37] Marcelo R Nascimento, Christian E Rothenberg, Marcos R Salvador, Carlos NA Corrêa, Sidney C de Lucena, and Maurício F Magalhães. Virtual routers as a service: the routeflow approach leveraging software-defined networks. In *Proc. CFI*, pages 34–37. ACM, 2011.

[38] T. Nguyen and G. Armitage. A survey of techniques for Internet traffic classification using machine learning. *IEEE communications surveys and tutorials*, 10(4), 2008.

[39] I. Pepelnjak. Could IXPs Use OpenFlow To Scale?, 2013.

[40] Peter Phaal, Sonia Panchen, and Neil McKee. Inmon corporation's sflow: A method for monitoring traffic in switched and routed networks. Technical report, RFC 3176, 2001.

[41] K. Phemius and M Bouet. Monitoring latency with OpenFlow. In *Proc. CNSM*, pages 122–125, 2013.

[42] I. Poese, B. Frank, B. Ager, G. Smaragdakis, S. Uhlig, and A. Feldmann. Improving Content Delivery with PaDIS. *IEEE Internet Computing*, 2012.

[43] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion Detection with Unlabeled Data Using Clustering. In *DMSA Workshop*. ACM, 2001.

[44] B. Quoitin, C. Pelsser, O. Bonaventure, and S. Uhlig. A Performance Evaluation of BGP-based Traffic Engineering. *International Journal of Network Management*, 15(3):177–191, 2005.

[45] Bruno Quoitin, Cristel Pelsser, Louis Swinnen, Olivier Bonaventure, and Steve Uhlig. Interdomain Traffic Engineering with BGP. *IEEE Communication Magazine*, 41(5):122–128, 2003.

[46] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, and Mark Handley. Improving datacenter performance and robustness with multipath tcp. *ACM CCR*, 41(4):266–277, 2011.

[47] M. Roesch. Snort: Lightweight Intrusion Detection for Networks. In *13th Systems Administration Conference*. USENIX, 1999.

[48] A Sajassi, R Aggarwal, J Uttaro, N Bitar, W Henderickx, and A Isaac. Requirements for ethernet vpn (evpn). Technical report, 2014.

[49] Michael Scharf and Alan Ford. Multipath tcp (mptcp) application interface considerations. Technical report, 2013.

[50] Anees Shaikh, Renu Tewari, and Mukesh Agrawal. On the Effectiveness of DNS-based Server Selection. In *Proc. INFOCOM*, volume 3, pages 1801–1810. IEEE, 2001.

[51] A. Soule, K. Salamatian, and N. Taft. Combining Filtering and Statistical Methods for Anomaly Detection. In *International Measurement Conference*. ACM, 2005.

[52] M. Sridharan, K. Duda, I. Ganga, A. Greenberg, G. Lin, M. Pearson, P. Thaler, C. Tumuluri, N. Venkataramaiah, and Y. Wang. NVGRE: Network Virtualization using Generic Routing Encapsulation. Internet draft, Internet Engineering Task Force, Sep 2011.

[53] M. Thottan and J. Chuanyi. Anomaly Detection in IP Networks. *IEEE Transaction on Signal Processing*, 51(8), 2003.

[54] Amin Tootoonchian, Monia Ghobadi, and Yashar Ganjali. Opentm: traffic matrix estimator for openflow networks. In *Proc. PAM*, pages 201–210. Springer, 2010.

[55] Hiroshi Tsunoda and Glenn Mansfield Keeni. Security by simple network traffic monitoring. In *Proc SIN*, SIN '12, pages 201–204, New York, NY, USA, 2012. ACM.

[56] S. Uhlig and O. Bonaventure. Designing BGP-based Outbound Traffic Engineering Techniques for Stub ASes. *ACM CCR*, 34(5):89–106, Oct 2004.

[57] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon. Providing Public Intradomain Traffic Matrices to the Research Community. *ACM CCR*, 36(1), 2006.

[58] Niels LM Van Adrichem, Christian Doerr, Fernando Kuipers, et al. Opennetmon: Network monitoring in openflow software-defined networks. In *Proc. NOMS*, pages 1–8. IEEE, 2014.

[59] Mea Wang, Baochun Li, and Zongpeng Li. sflow: Towards resource-efficient and agile service federation in service overlay networks. In *Proc. IEEE ICDCS*, pages 628–635. IEEE, 2004.

[60] JonathanStuart Ward and Adam Barker. Observing the clouds: a survey and taxonomy of cloud monitoring. *Journal of Cloud Computing*, 3(1), 2014.

[61] Kuai Xu, Feng Wang, and Haiyan Wang. Lightweight and informative traffic metrics for data center monitoring. *J. Netw. Syst. Manage.*, 20(2):226–243, Jun 2012.

[62] Kuai Xu, Zhi-Li Zhang, and Supratik Bhattacharyya. Internet traffic behavior profiling for network security monitoring. *IEEE/ACM Trans. Networking*, 16(6):1241–1252, Dec 2008.

[63] Curtis Yu, Cristian Lumezanu, Yueping Zhang, Vishal Singh, Guofei Jiang, and Harsha V Madhyastha. Flowsense: Monitoring network utilization with zero measurement cost. In *Proc. PAM*, pages 31–41. Springer, 2013.

[64] Qi Zhao, Zihui Ge, Jia Wang, and Jun Xu. Robust Traffic Matrix Estimation with Imperfect Information: Making Use of Multiple Data Sources. *SIGMETRICS Perform. Eval. Rev.*, 34(1):133–144, 2006.