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
Document version	0.6
Editor in Chief	Castro, QMUL
Author	Castro, Fernandes, Antichi, Gusat, Kathareios, Uhlig
Date	15/12/2015
Reviewers	DE-CIX
Date of review	07/12/2015
Status	*Public*

Revision History

Date	Version	Description	Author
03/07/15	0.1	First draft	Castro, QMUL; Fernandes, QMUL; Antichi, UCAM; Gusat, IBM; Kathareios, IBM
03/07/15	0.2	Minor changes	Uhlig, QMUL
25/11/15	0.3	Table & new references	Castro, QMUL
25/11/15	0.4	Review	Dietzel, DE-CIX; Bleidner, DE-CIX
25/11/15	0.5	Reviewer requested changes	Castro, QMUL
15/12/15	0.6	Final version	Castro, QMUL

Executive summary

The declared mission of ENDEAVOUR is to advance the Internet interconnection model to a new paradigm through the introduction of Software Defined Network (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.

Contents

List of Tables

1 Introduction

The deployment of SDN at IXPs leads to new monitoring requirements, to meet the potentially stringent requirements posed by the Service-Level Agreement (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.

Use cases	Monitoring requirements	
	Control plane	Data plane
Load Balancing	x	traffic volume per physical port
Traffic engineering	x	traffic volume per physical port
Peering	control/data plane consistency	traffic volume per physical port
Overlay	changes in routing entries	FIB
Security	control/data plane consistency	contingent on the specific security aspect
Traffic steering	BGP announcements	traffic volume per physical port
Routing As a Service (RAS)	-FIB (routes) -convergence time	-consistency with FIB -topology

Table 1: Monitoring requirements per use case

To better understand monitoring requirements and its challenges, we analyse these two issues separately. First we examine the monitoring re-

quirements 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 Autonomous Systems (ASes). Table 1 provides a summarized view of the key monitoring requirements for each use case. While we used the use cases and whether the specific monitoring requirement affects the control or the data plane as a classification criteria, other classifications are possible. In particular we will explore in future deliverables the use of the *Volere* specification methodology for requirements [51].

After analysing the monitoring requirements, we examine the challenges that they imply. 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 [54]. 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 [33] to overcome the limitations of content-aware traffic engineering based on DNS tweaking [30, 46]. 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 (LPS) and Virtual Private LAN Services (VPLS).

Monitoring the fabric data plane is fundamental to ensure that the chosen load balancing policy is adequately implemented [33]. 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

Border Gateway Protocol (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 ASes [47, 16, 48], the performance of these techniques is very limited [33]. 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 [60]. 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, SDN's 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 [33]. 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 OVNs 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 OVNs 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. Due to privacy concerns new way to encrypt and monitor the data plane are worth considering [15].

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

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

example is the possibility to prevent the Address Resolution Protocol (ARP) 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 [43]. 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 [11].

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 OpenFlow-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.

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)

RAS was a SDN predecessor, which already presented a clear separation of the control and data planes [38]. 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 [42]. 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 Inter-domain 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 Open Shortest Path First (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:

- **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.
- **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).

- **Security:** this class of monitoring assists in attack, intrusion and DDoS 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:

- **Local**: typically performed at the core of the network, at each graph vertex, it measures queue occupancy (port, link or interface).
- **Path**: performed at the edge of the network, it measures throughput and latency on one or more graph paths.
- **Global**: more recent methods aim at global monitoring, measuring all the components of the network graph, creating congestion matrices, heatmaps [6].

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

- **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)
- **Packet capture (Pcap)**: performed at one core vertex or edge by capturing traversing packets (incoming and outgoing).
- **Probing/Telemetry**: Performed edge-to-edge, aims at collecting statistics for the interconnecting path such as Round Trip Time (RTT) or drop rate.
- **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 [44], NetFlow [19] and SNMP [13].

While traffic matrix-based approaches facilitate decision making based on information gathered from the monitoring and measurements within the network in sFlow [63], IPFIX [18], or Netflow [20, 27] 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 OpenFlow and OpenStack offer new monitoring capabilities and APIs (e.g., richer per flow state, new counters and statistics, etc.) [45], SDN introduces new elements (overlays, tunnels, hypervisors and container/dockers, vswitches and vNICs) whose monitoring proves to be challenging [8, 23].

While traffic matrices are crucial for capacity planning, traffic engineering and routing protocol configuration [61, 58], traffic matrix estimation is problematic [68]. To address the limitations of current traffic matrix estimation methods in SDNs, new proposals such as [67, 26, 62, 32] have emerged.

3.2.3 Cloud monitoring

Cloud monitoring is crucial [64] to accurately quantify the performance provided by the infrastructure. While new monitoring techniques are being continuously proposed [37, 9, 36, 35, 6], these techniques typically face significant challenges with regard to scale, rapidity, detection, localization, and diagnose of performance problems [29].

3.2.4 Overlay monitoring

The popularity of tunnels and overlays (for instance to implement remote peering [14]) 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 [64, 65, 6].

With an overwhelming variety of virtualization techniques [17, 39, 55, 31, 12, 41], VXLAN is emerging as the de-facto standard for the future of SDN-based OVNs/tunneling [3] for datacenter networks.

While scalability remains limited, an emerging Internet Engineering Task Force (IETF) standard [34, 52] that uses MP-BGP for MAC learning in

the control plane addresses the problem and extends the VXLAN across a WAN [3]. 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 Transport Control Protocol (TCP)/User Datagram Protocol (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) [5], 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) [49, 53] is another modification of TCP. MPTCP can outperform TCP [49] by offering path redundancy through the simultaneous use of several IP-addresses/interfaces. The Fast and Secure Protocol (FASP) [28] overcomes the performance bottleneck of TCP when moving massive data, particularly for WANs with large bandwidth, high round-trip time and packet loss. zFabric [22], an SDN-based tunnelling transport mechanism built on zOVN [23], 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

Attack and intrusion detection, DDoS detection, Firewall (h/w or as NFV) [66] 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 [59] while still providing a lightweight approach [65].

4 Monitoring capabilities of contemporary switches

Continuing on the analysis of commercially-available switches described in [2], work is being performed towards evaluating the pros and cons of their moni-

toring capabilities, with respect to the specific monitoring requirements that we envision for the ENDEAVOUR ecosystem.

Most switches rely on SNMP polling [13], sFlow [44], and the OpenFlow object database [40] for monitoring. These tools have their measurement data going through the switch CPU, and thus are limited in speed by the CPU's overall capabilities and load at the moment that the process takes place. For example, OpenFlow provides two measurement techniques to monitor the network: *packet_in* messages and flow counters. In practice though, neither of these mechanisms are either scalable or provide low-latency measurements. The *packet_in* messages are limited to at most a few hundreds per second [56, 25] and they only provide data when a flow first appears or expires. Port and flow counters on the other hand are typically only updated every second [26], which is too slow to identify even medium-sized flows [10]. sFlow is similarly constricted by the CPU performance [57, 50]. Increasing the packet sampling rate applies extra burden on the switch CPU, and has been shown to peak at 300 or 350 samples per second on state-of-the-art switches.

The Brocade, Mellanox, and Arista switches (see [2]) offer port mirroring as an alternative. Port mirroring is a mechanism that copies all packets flowing through a subset of the switch's ports to a monitoring port for telemetry and security purposes. Bypassing the switch CPU in the path of the monitoring data, grants monitoring techniques based on port mirroring a better performance, faster than sFlow-based techniques by a factor of 10 [50]. However, such methods have not been tested at higher network scales and line rates, where an impactful overhead of monitoring ports per switch is required.

More integrated monitoring solutions are implemented in various switch operating systems. As examples, Broadcom's BroadView [21], Arista CloudVision [7], and Cumulus Linux [24] (available for a wide range of commercial hardware) promise real-time reporting of measurements such as queue-length and buffer utilization.

Finally, it should be noted that the Broadcom Trident II is the only switch that implements high-speed packet and queue sampling through the data-path, although it is often not enabled on COTS switches that use the chipset (as is the case of Arista switches). This is performed by the IEEE 802.1Qau standard, Quantized Congestion Notification (QCN) [4], a suitable candidate for microsecond scale port monitoring through the data-path, that bypasses both the limitations of the switch CPU and the scalability issues of port mirroring.

5 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.

6 Acronyms

SDN Software Defined Network

BGP Border Gateway Protocol

IXP Internet eXchange Point

SLA Service-Level Agreement

AS Autonomous System

RAS Routing As a Service

DDoS Distributed Denial of Service

DNS Domain Name System

TTL Time To Leave

LPS Label Path Switching

VPLS Virtual Private LAN Services

OVN Overlay Virtual Network

DPI Deep Packet Inspection

FIB Forwarding Information Base

VM Virtual Machine

RTT Round Trip Time

IETF Internet Engineering Task Force

DCTCP Congestion Datacenter TCP

MPTCP Multi-Path TCP

FASP Fast and Secure Protocol

TCP Transport Control Protocol

UDP User Datagram Protocol

QCN Quantized Congestion Notification

ARP Address Resolution Protocol

OSPF Open Shortest Path First

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] D.2.1: Requirements and initial design.

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

[4] Mohammad Alizadeh, Berk Atikoglu, Abdul Kabbani, Ashvin Lakshmikantha, Rong Pan, Balaji Prabhakar, and Mick Seaman. Data center transport mechanisms: Congestion control theory and ieee standardization. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 1270–1277. IEEE, 2008.

[5] 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.

[6] 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.

[7] Arista. Arista EOS CloudVision: Cloud Automation for Everyone. `https://www.arista.com/assets/data/pdf/Whitepapers/CloudVision_WP_0815.pdf`.

[8] 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.

[9] 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.

[10] Theophilus Benson, Ashok Anand, Aditya Akella, and Ming Zhang. Microte: Fine grained traffic engineering for data centers. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, page 8. ACM, 2011.

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

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

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

[14] Ignacio Castro, Juan Camilo Cardona, Sergey Gorinsky, and Pierre Francois. Remote Peering: More Peering without Internet Flattening. In *Proceedings of CoNEXT*. ACM, 2014.

[15] Ignacio Castro, Aurojit Panda, Barath Raghavan, Scott Shenker, and Sergey Gorinsky. Route Bazaar: Automatic Interdomain Contract Negotiation. In *Proceedings of HotOS*. USENIX Association, 2015.

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

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

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

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

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

[21] Broadcom Corp. Building an Open Source Data Center Monitoring Tool Using Broadcom BroadView Instrumentation Software. `https://www.broadcom.com/collateral/tb/BroadView-TB200-R.pdf`.

[22] 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.

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

[24] Cumulus. Cumulus Linux, The first, true Linux OS for data center networking. `https://cumulusnetworks.com/media/cumulus/pdf/misc/Cumulus-Linux-Datasheet.pdf`.

[25] Andrew R Curtis, Jeffrey C Mogul, Jean Tourrilhes, Praveen Yalagandula, Puneet Sharma, and Sujata Banerjee. Devoflow: Scaling flow management for high-performance networks. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 254–265. ACM, 2011.

[26] 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.

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

[28] 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.

[29] 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.

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

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

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

[40] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, , and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM CCR*, 38(2):69–74, Apr 2008.

[41] 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.

[42] 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.

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

[44] 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.

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

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

[47] 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.

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

[49] 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.

[50] Jeff Rasley, Brent Stephens, Colin Dixon, Eric Rozner, Wes Felter, Kanak Agarwal, John Carter, and Rodrigo Fonseca. Planck: millisecond-scale monitoring and control for commodity networks. In *Proceedings of the 2014 ACM conference on SIGCOMM*, pages 407–418. ACM, 2014.

[51] James Robertson and S Robertson. Volere. *Requirements Specification Templates*, 2015.

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

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

[54] 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.

[55] 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.

[56] Brent Stephens, Alan Cox, Wes Felter, Colin Dixon, and John Carter. Past: Scalable ethernet for data centers. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 49–60. ACM, 2012.

[57] Junho Suh, TT Kwon, C Dixon, W Felter, and J Carter. Opensample: A low-latency, sampling-based measurement platform for sdn. ICDCS, 2014.

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

[59] 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.

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

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

[62] 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.

[63] 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.

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

[65] 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.

[66] 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.

[67] 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.

[68] 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.