This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

Software Defined Management of Edge as a Service Networks

Rafael L. Gomes, *Member, IEEE,* Luiz F. Bittencourt, *Member, IEEE,* Edmundo R. M. Madeira, *Member, IEEE,* Eduardo C. Cerqueira, *Member, IEEE,* and Mario Gerla, *Fellow, IEEE*

***Abstract*—It is a consensus that the Internet suffers from architectural limitations, including resilience, scalability, and manageability, among others. Therefore, companies access the Internet by establishing a Service Level Agreement (SLA), in the attempt to ensure Quality of Service (QoS) for users. To address the current limitations of the Internet, researchers have recently proposed the Edge as a Service (EaaS) paradigm as a suitable solution to improve the access capacity of edge networks. EaaS uses Network Virtualization and Software Defined Networks to expand flexibility and manageability of access to edge network resources. Moreover, to maintain QoS assurance for users, EaaS addresses network events (such as traffic overload, failures, etc.) that can potentially affect QoS. Within this context, this article proposes a Software Defined Management of EaaS environment, called SDM-EaaS. The proposal enhances the QoS for the end user, while improving the utilization of network resources in dynamic scenarios allowing for unpredictable changes in traffic demands, network infrastructure availability and customer characteristics. Experiments based on emulation as well as real testbed demonstrate the effectiveness of the SDM-EaaS strategy.**

***Index Terms*—Virtual Networks, Software Defined Network, Service Level Agreement, Edge as a Service.**

I. Introduction

Currently, the Internet is the primary medium for content sharing, playing a central role in user lifestyles. As users are becoming mobile, wireless access to content has dramatically increased in the past few years. This trend has enormously amplified resource demands in edge and access networks. Moreover, unpredictable mobility creates a new paradigm where access to content can be expected anytime, anywhere and with best effort quality level. Users become frustrated when they see the quality level of their services decreases. This new dynamic traffic demand paradigm requires an elastic allocation of networking resources to achieve adequate Quality of Service (QoS) assurance for Internet services. Moreover, the network must be resilient enough to withstand traffic overload and equipment faults during normal operation [1].

According to Davy et al. [2], to address this new paradigm, Internet Service Providers (ISPs) tend to apply the Edge-as-a-Service (EaaS) approach. The EaaS concept allows the implementation of an elastic Virtual Network Resource (VNR) approach [2]. When coupled with the Service Level Agreement (SLA), this new scheme creates a modus operandi in which the user agrees on a basic SLA contract, but also specifies SLA adjustments to be made according to client defined policies. In this article, the term "client" refers to the Virtual Access Network, and the term "user" represents the final user that accesses the Internet to upload/download content.

EaaS employs Network Virtualization (NV) and Software Defined Network (SDN) principles to allocate edge network resources with flexibility and to make the network service adaptable/customizable [2]. NV is a technology that allows the deployment, over a single physical infrastructure, of multiple network slices with customized, dynamically adjusted properties, corresponding to the required Virtual Network (VN) behavior. The usage of VNs is viewed as a means for network operators to cope with both the varying demand for high-bandwidth services and the lack of flexibility in current network management. Similarly, at a lower layer, SDN is being exploited by network operators to reduce the load on network components and to enable a more effective usage of elastic resources [3]. SDN and NV controls can be integrated in a Network Hypervisor (such as Flowvisor [4] or OpenVirtex [5]), which allows the slicing of the network in layers. Each layer has a particular set of resources and protocols, corresponding to a customized VN. We call this a *Virtual Software Defined Network* (VSDN).

The management of virtual networks under EaaS environments is still an open issue, since it demands [2]: (i) support to SLAs; (ii) re-distribution of VNR; (iii) mediation of competing requests for VNR; (iv) allocation schemes that consider access technology and economic value; and, (v) management policy sets. Additionally, many types of network events (e.g., traffic overload, failures, etc.) can occur, affecting both the QoS experienced by the user and the SLA definition. The EaaS exploits the advantages of SDN for management issues [6]. More precisely: (i) with SDN it is easier to modify the behavior of the network and; (ii) the SDN centralized view allows global, up to date knowledge of network state. With SDN global state, it is possible to identify the network events that will affect QoS and thus SLA [6]. Moreover, exploiting the VN capabilities, the VSDN architecture enables the isolation and customization of network behavior through individual controllers. Namely, it is possible to customize a specific VSDN without changing the other VSDNs.

Within this context, this article proposes SDM-EaaS, a Software Defined Management strategy of VSDNs in an EaaS environment, dealing with the features claimed by EaaS in an integrated way. One of the important functions of SDM is to identify network events and adapt VSDN characteristics (such as resources allocated, network topology, to name a

R. L. Gomes., L. F. Bittencourt, and E. R. M. Madeira are with University of Campinas (UNICAMP).

E. Cerqueira is with Federal University of Para (UFPA).

M. Gerla is with University of California, Los Angeles (UCLA).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

few) to address each type of network event. Three types of network events are addressed: (1) Adjustment; (2) Failure; and (3) Scheduling. These events are detailed in the sequel, and are addressed by specific algorithms and mechanisms. Finally, the SDM-EaaS architecture is evaluated through emulation and real testbed experiments, yielding improvements over EaaS.

This article is organized as follows. Section II presents related work, encompassing topics related to management of resource, VNs and/or SDNs. Section III describes the proposed SDM-EaaS architecture, while Section IV details the occurrence of network events. Section V shows the experiments performed to evaluate the SDM-EaaS architecture and the proposed mechanisms to address the network events. Section VI summarizes the article and presents some future work.

II. Related Work

This section describes key related work about resource management strategies of SDN and VN environments. Table I summarizes these existing work in the literature, highlighting the differences to our proposal in relation to the key issues claimed by the EaaS [2]: (i) support to SLAs; (ii) elastic allocation of resources; (iii) sharing of VNR for competing requests; (iv) schemes that focus on access technology and economy; and, (v) usage of management policy sets.

Mijumbi et al. [7] propose a solution to the unsplittable flow virtual network embedding problem. The authors perform both virtual nodes and links embedding in one step using mathematical programming and path generation, applying price as a criterion. However, this paper does not use SDN and does not address the network events that could affect the virtual networks. Additionally, this paper cannot cope with service, user and network dynamics aspects claimed by EaaS.

Grossglauser et al. [8] present an admission control based on traffic measurements to learn the statistics of the existing flows, with that the authors aim to avoid overallocation of network resources and the admission control does not need a traffic description. Individually, this approach can not deal with the elastic behavior demanded by EaaS. On the other hand, this approach could be attached in a future work to the SDM-EaaS architecture to work together with an adjustment mechanism, being applied when an adjustment can not be performed anymore due to lack of resources.

Song et al. [9] propose a globally deployable Network Embedded On-Line Disaster (NEOD) management framework for SDNs. NEOD is designed as an extension of Openflow, focusing on real time identification of network disaster events (multiple failures in the same region) through embedded event detectors. Despite dealing with failures, it does not handle other types of event, as well as it does not consider network virtualization issues, impacting directly in the re-distribution of virtual network resources among the clients when a failure occurs, and consequently in the service delivery by EaaS.

Kim et al. [6] analyze how the SDN principles can improve the management and configuration of computer networks. Moreover, the authors propose PROCERA, a network control framework that helps operators to express configuration policies that react to conditions such as user authentication, time of the day, bandwidth usage, or server load. PROCERA is a reactive framework, because it uses many realistic network policies to react to dynamic changes in network conditions. However, PROCERA has not been designed to deal with virtual networks and it does not consider the network events that can affect QoS and SLA issues.

Lo et al. [10] present a framework to provide dynamic bandwidth adjustment, called Flexible Network Management Framework based on OpenFlow (FNMF-OF). This framework supports traffic statistics and shares underutilized network resources to satisfy high priority users profiles and service requirements. FNMF-OF focuses on networks that have access to the information of users and applications to make the profiles for resource sharing. Hence, it was not designed to maintain and dynamically grow/shrink virtual networks, as well as to mediate competing requests for virtual resources. These facts disable the deployment of FNMF-OF in EaaS.

Tuncer et al. [11] proposed an SDN-based management and control framework for fixed backbone networks, which provides support for both static and dynamic resource management applications. The exchange of information between distributed elements of the framework is supported by the proposed management substrate. This fact restricts the usage of the framework to modify the existing SDN equipment. Additionally, this framework does not consider network virtualization and network events. Moreover, the EaaS needs to address the SLA for short lived virtual networks.

Basta et al. [12] show four openflow-based architectures to deal with Network Function Virtualization (NFV) under LTE scenarios. The proposal deploys an extra Openflow element, called NE+, to add new network functions to the LTE environment. However, this architecture focuses on LTE functions and NFV, without acting in the virtualization environment directly. This lack of virtual resource management limits the usage of this architecture in EaaS since no dynamic allocation is provided and it does not consider event handling.

Lin et al. [13] present Software Defined Infrastructure (SDI) to perform an integrated control and management of computing and network resources. SDI applies a cloud controller and an SDN controller, where the exchange of information between them is used to manage the network resources. Nevertheless, the EaaS does not have an entity which accesses the application behavior, consequently this lack of specific information hinders the usage of SDI in EaaS.

Nguyen et al. [14] show an SDN framework designed for embedding multi-level virtual networks in physical infrastructures, called ReServNet. The ReServNet framework focuses on a hierarchical link sharing and bandwidth reservation for the virtual networks. However, the ReServNet framework does not perform resource adjustment for the virtual network, as well as it does not consider network events. These facts express the inadequacy of the ReServNet to the EaaS, since EaaS needs to maintain and dynamically grow/shrink virtual networks.

He et al. [15] show DaVinci architecture to periodically reassign bandwidth in each substrate link between its virtual links. In the DaVinci architecture, each virtual network runs a distributed protocol that maximizes its own performance objective independently, considering that CPU resources have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

TABLE I
RELATED WORK

Reference	Network	SLA Support	Elastic allocation	VNR Sharing	Policy-Aware	Events
Mijumbi et al. [7]	VN	No	No	Yes	No	No
Grossglauser et al. [8]	VN	No	No	Yes	No	No
Song et al. [9]	SDN	No	No	No	No	Failures only
Kim et al. [6]	SDN	No	No	No	Yes	No
Lo et al. [10]	SDN	No	Yes	No	No	No
Tuncer et al. [11]	SDN	No	Yes	No	No	No
Basta et al. [12]	SDN and LTE	Yes	No	No	No	No
Lin et al. [13]	SDN and Cloud	Yes	No	Yes	No	No
Nguyen et al. [14]	SDN and VN	No	No	Yes	No	No
He et al. [15]	SDN and VN	No	No	Yes	No	No
Bueno et al. [16]	SDN and VN	No	No	Yes	No	No
Hongyun et al. [17]	SDN and VN	Fixed services	No	Yes	No	No
SDM-EaaS	SDN and VN	Yes	Yes	Yes	Yes	Yes

the most direct impact on packet-delivery services. Hence, it is hard to address the elastic behavior demanded by the EaaS, since DaVinci does not consider the adjustment of virtual network resources based on the network infrastructure as a whole. Additionally, DaVinci was not designed to support network programmability, complicating its usage in EaaS.

Bueno et al. [16] propose a network control layer for heterogeneous infrastructures based on SDN and Network as a Service paradigms. It aims to perform a flow-oriented and on demand adjustment of network resources upon changes in the application requirements, i.e., this proposal needs the information coming from the user application. This fact makes it inappropriate to EaaS, which does not have direct interaction with the user. Furthermore, this proposal does not deal with virtual network environments and it does not have an event-aware behavior.

Hongyun et al. [17] present a management architecture for multi-service under SDN, called AMA. AMA architecture applies a VN approach to isolate each type services, where the following three types of services are considered: voice, video, and non real-time Internet service. This fact limits the application in EaaS, which needs a flexible scheme to address the dynamic of the scenario. Additionally, two points prevent AMA to be used in EaaS: (a) AMA does not allow on demand redistribution of virtualized network resources; and, it considers a previous informed requirements of the services as base for VN deployment instead of an SLA.

To the best of our knowledge, there are no works in the literature that focus on the development of an architecture to deal with the all features demanded by the EaaS. These features are integrated and addressed in the architecture described in this article.

III. SDM-EAAS ARCHITECTURE

The EaaS enables service providers to use virtualization and build dedicated, elastic virtual network, tailored to reach their customers [2]. EaaS decouples the strict ownership relationship between network operators and their access networks. Through the development of virtual networks, adaptative functions can be requested on-demand to support the delivery of more flexible services. In EaaS, the main interest lies on the behavior of the aggregated users and the traffic volume generated. Figure 1 illustrates the EaaS scenario addressed in this article. The user is connected to an access network (the client) that is linked to the Internet by an edge network. In this article, the term "client" refers to the access network (i.e., the ISP's client, which signs the SLA), and the term "user" is the final user. The final user accesses the Internet to upload/download content and run applications, for example using mobile phones, tablets, notebooks, and others.

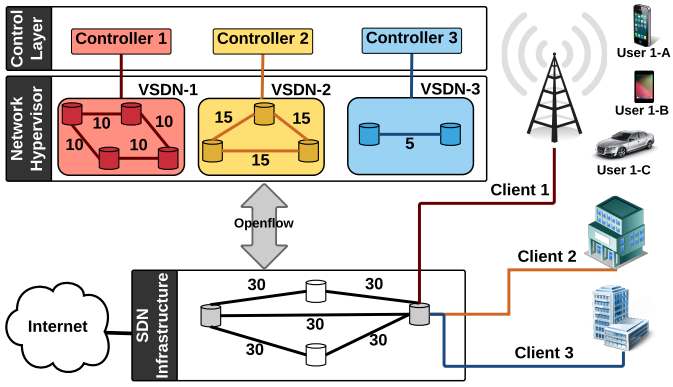

Fig. 1. Scenario representing the context of the proposed SDM-EaaS.

Based on Figure 1, an example of usage can be described as follows: if the client deploys a VSDN with an ISP, it can allocate the VSDN-1 with 10 Mpbs, which has $Controller1$. Independently, the ISP could deploy the VSDN-2 with 15 Mpbs and VSDN-3 with 5 Mbps, that have $Controller2$ and $Controller3$, respectively. The service delivered to each client is different (since, the resource and the behavior of the VSDN could be distinct) and they are isolated among them. However, the users of the same client share the resources allocated to the client.

When a user starts a flow, the SDN switch detects that this flow does not match any existing flow on its flow table. Thus, the SDN switch sends the flow information to the network hypervisor, that checks which controller is responsible for this flow (i.e., it checks which virtual network this flow is part of) and forwards this information to the controller. The controller analyzes this information and decides what action the SDN switch must take with this flow. The controller knows only

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

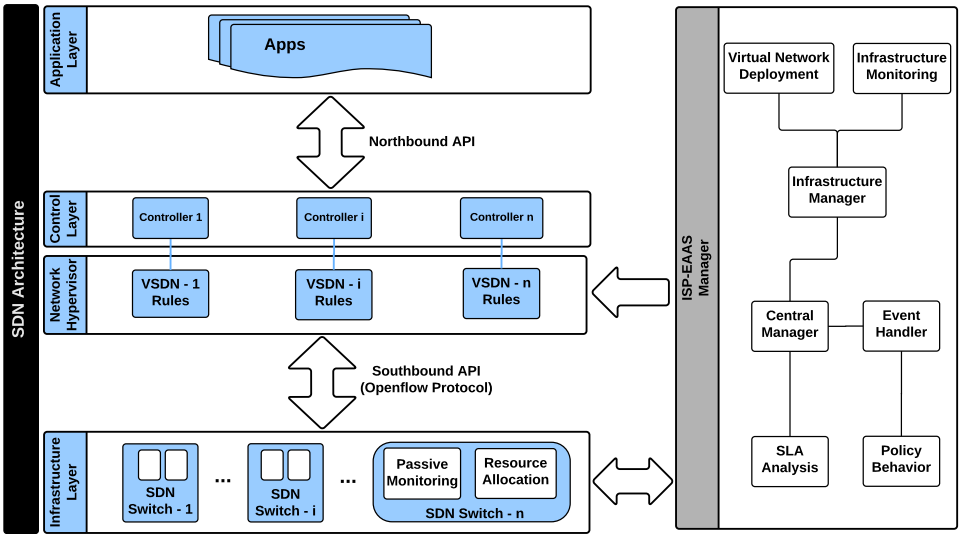

Fig. 2. Overview of the SDM-EaaS architecture.

the network components and resources allocated to it.

The binding of a flow to a virtual network depends on the ISP's desire, which could be based on many criteria, like flow characteristics, SLA definition, among other. In the SDN switch, this binding represents that a flow is linked to a specific virtual queue for its virtual network, which controls the resource utilized by the flows of the virtual network.

Within this context, it is possible to manage each VSDN in a separate way according to the preferences defined by the client in the SLA definition. For example, when VSDN-1 identifies a congestion period, the ISP can increase the amount of resources allocated to it. On the other hand, the ISP could address the congestion in a different way for VSDN-2, applying an admission control over new flows. Therefore, the behavior of each VSDN can be adapted without influencing other VSDNs under the same physical infrastructure.

Measurements from real networks [18] indicate that the traffic demand fluctuates over time. Hence, a static resource allocation scheme can be inefficient, thus degrading user experience. Traffic variability is highly relevant to the EaaS. Users access high volumes of data and video content, and resource demands vary over time [2]. Similarly, failures in the network (links and nodes) can compromise the already established SLA, as well as impact the QoS experienced by the user.

The SDM-EaaS architecture focuses on the management of EaaS, which claims for solutions that encompass [2]: (i) faster and flexible support to SLAs, facilitating very short lived virtual networks; (ii) techniques to maintain and dynamically grow/shrink virtual networks, re-distributing virtualized network resources; (iii) solutions to mediate many potentially competing requests for virtualized resources, for instance, to maximize the resource utilization; (iv) resource allocation schemes and policies that take into account both access technology and economic value; and (v) analysis and consistency checking of management policy sets.

To accomplish the above, the proposed SDM-EaaS architecture changes the VSDN characteristics according to defined policies, addressing each type of network event. Consistent with SDN principles, SDM-EaaS allows the network administrator to associate with each VSDN the policies to be followed when different network events occur. Thus, our solution allows the customization of network parameters and services, providing flexibility to adapt the service according to the clients requirements. It can guarantee the SLA contract and, at the same time, it isolates the behavior of each VSDN. The idea is to give flexibility and automatic management to the EaaS environment, providing a suitable QoS for the user, while keeping the SLA required by the client.

Figure 2 illustrates the SDM-EaaS architecture and relationship between its modules. Each environment that exists in the EaaS context is represented by blue boxes and the modules of SDM-EaaS are depicted by white boxes. The whole environment has the following components: (a) Infrastructure Layer; (b) Network Hypervisor; (c) Control Layer; (d) Application Layer; and (e) ISP-EaaS Manager.

The SDM-EaaS architecture was designed to complement and extend the interaction between Controller and switches, i.e., SDM-EaaS attaches management capacity to the existing approaches of virtualization under SDN infrastructures. The *ISP-EaaS Manager* environment represents the central part of the SDM-EaaS architecture, which exchanges information between the *Infrastructure Layer*, the *Network Hypervisor* and the *VSDN Controllers*. Next, each module will be described, emphasizing their functionalities and their role in the SDM-EaaS architecture as a whole.

A. Infrastructure Layer

This section describes the modules related to the infrastructure environment, i.e., the set of SDN switches that compose the EaaS environment. Each SDN switch has an agent that encompasses the modules presented in the next sections.

1) Passive Monitoring: The SDM-EaaS architecture uses the infrastructure information to manage the EaaS environment, where the *Passive Monitoring* performs information gathering from each SDN switch. Basically, it is aware of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

active flows in the switch, the traffic volume of each flow, the switching information and the active ports.

2) Resource Allocation: After the *ISP-EaaS Manager* evaluates the EaaS condition and decides to adjust a VSDN, it sends a message to a set of switches with the actions to be performed. Thus, the *Resource Allocation* module receives the message and performs the resource allocation for the VSDN on-the-fly.

B. ISP-EaaS Manager

This section details the modules regarding the *ISP-EaaS Manager*. It is the main part of the SDM-EaaS architecture, since it monitors the environment, makes decisions, manages SLA definitions, addresses network events and triggers policies.

1) Central Manager: The ISP-EaaS Manager needs a central module to control the interaction among the other modules, i.e., the behavior of the architecture as a whole. In this way, the *Central Manager* module manages the service provision, the virtual network deployment and the management of resources within a VN.

First, it defines and negotiates the SLA. Second, it deploys the VSDN. After that, it uses the other modules to perform the monitoring and to manage the EaaS environment when a network event is detected.

2) SLA Analysis: To manage the EaaS environment, the SDM-EaaS architecture takes into account the current SLAs, considering a faster and flexible support to SLAs, which is the issue (i) claimed by the EaaS (faster and flexible support to SLAs) [2]. This task is performed by the *SLA Analysis* module. In the same way, it determines if it is feasible to deploy the VSDN requested by the client based on the SLA definition. It is not the focus of this paper to specify and to negotiate the SLA. We consider that this process is performed previously by another solution, for example the negotiation protocol presented in reference [19].

The *SLA Analysis* module can also initiate the deployment of a new VSDN when there are available resources to accept the request. Thus, this module checks the information about the resources available in the *Infrastructure Manager* and requests the resource allocation for the deployment of the new VSDN. In the same way, the *SLA Analysis* informs the *Virtual Network Deployment* module about the set of protocols defined in the SLA and instantiated in the VSDN. It configures the policy to be associated with each network event in the *Policy Behavior* module.

3) Infrastructure Manager: The deployment and the adjustment of the VSDN are based on the information about the available resources and network components in the SDN infrastructure. To this end, the *Infrastructure Manager* module is defined. This module interacts directly with the *Infrastructure Monitoring* and *Virtual Network Deployment* modules to control the resource information (availability) and to deploy (or adjust) the internal characteristics (network components allocation and protocols) of each VSDN, respectively.

More specifically, the *Infrastructure Manager* module acts as the ISP interface to resource management. It provides information about the network infrastructure, performs the adjustments, and controls the interactions between these tasks, which are all correlated. To perform the adjustment it is necessary to verify the available resources; on the other hand, when the resources are adjusted, the module needs to update the resource information regarding the infrastructure and the SLA of the VSDN. Hence, the tasks performed by the *Infrastructure Manager* module encompass the issues (ii) and (iii) claimed by the EaaS (management of both elasticity and competing requests in VN) [2].

4) Infrastructure Monitoring: The *Infrastructure Monitoring* module stores information regarding the resources and network components (links and nodes) that are available under the SDN infrastructure, which is different from the controller's perspective (which just sees the infrastructure allocated to it). Besides that, it keeps information about the resources and network components allocated to each VSDN, allowing the ISP to know if it is possible to adjust the resources of a specific VSDN. These data are used to perform the VSDN allocation, as well as to supply the policies.

The *Infrastructure Monitoring* module assists *Virtual Network Deployment* to perform the allocation and configuration of a VSDN. The information it provides is critical to allocate VSDNs on the EaaS environment.

5) Virtual Network Deployment: The main goals of the *Virtual Network Deployment* module are the following: (i) to generate the topology of the VSDN and configure it into the network hypervisor and (ii) to instantiate the set of protocols defined in the SLA under the SDN controller tied to the VSDN being deployed.

The topology generation of the VSDN aims to determine a network topology that is able to support the properties defined in the SLA, i.e., the switches and links to compose the VSDN. To perform this task, it is necessary to apply an allocation algorithm in the system, as the one presented in [20].

Once the network topology is defined, the *Virtual Network Deployment* is responsible for deploying it by communicating with the network hypervisor and informing the desired characteristics (network components and the set of protocols) in the VSDN.

6) Event Handler: Since the *Infrastructure Monitoring* module has the information about the EaaS environment, it is necessary to identify the network events and trigger the policy attached to address it. Thus, the *Event Handler* is responsible for these tasks, in which the network events are: (1) Adjustment; (2) Failure; and (3) Scheduling. More details about the context and the characteristics of each network event will be provided in Section IV.

The *Infrastructure Monitoring* module receives from other modules the following information: (i) traffic demand of each VSDN; (ii) the status (operational or failure) of each network component; and (iii) the initial SLA definition of each VSDN. In the Event Handler, each VSDN has a policy attached to each network event. In this way, when a network event occurs, the SDM-EaaS architecture allows the automatic reconfiguration of the EaaS environment to address it. The policy to be applied is specified in the SLA definition, i.e., it follows the client

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

requirements. Note that the policies attached to VSDNs can be different from one VSDN to the other.

Hence, when a network event is identified, the *Event Handler* module triggers the policy for the network event, which is defined on the *Policy Behavior* module.

7) Policy Behavior: Each policy addresses the network events in a different way, i.e., each one performs different procedures. For example, one VSDN could prefer to address congestion by deploying an admission control over new flows, while another VSDN could choose to request an increase in resource allocation.

To give the flexibility and the isolation for the VSDNs inside the EaaS environment, the SDM-EaaS architecture allows customized policies to different VSDNs via the *Policy Behavior* module. In this way, the VSDN can specify the desired policy. One important point regarding network policy definition is the consistency between policies, guaranteeing that one policy will not affect another one of the same or of a different VSDN. Hence, the SDM-EaaS architecture supports the issues (iv) and (v) claimed by the Eaas (resource allocation policies and their analysis) [2].

The specification and validation of the policy behavior, are not the focus of this article. A possible approach to be integrated to the SDM-EaaS architecture is the adaptation of one of the existing network policy management proposals [21], [22], [6] to formally characterize and validate policies. However, the SDM-EaaS architecture is open to attach the policies desired by the client (i.e. VSDN) whenever they are specified in the SLA.

C. Network Hypervisor, Control and Application Layers

This section presents the modules related to the VSDN environments, i.e., the set of SDN controllers and the set of rules to generate the VSDN that each controller is responsible for. Each controller has a set of Apps, which define the behavior of the VSDN (the deployed functionalities).

The *Control Layer* represents the existing SDN controllers that are deployed to control a VSDN, for example the Ryu Controller[1]. The *Application Layer* denotes the set of Apps that can be deployed in a controller. On the other hand, the *VSDN Rule* depicts the specification of the set of network components that will be allocated to a specific VSDN on the *Network Hypervisor*. All of them are entities that already exist to perform a VSDN deployment. Thus, they are just presented in Figure 2 to illustrate the interaction of SDM-EaaS architecture with the current approaches to have virtual networks under SDN infrastructures, which characterizes the EaaS environment.

IV. Network Events Definition

The SDM-EaaS architecture defines three types of network events that can occur in the EaaS environment: (1) *Congestion/Wastage*; (2) *Failure*; and (3) *Scheduling*. Each network event is described below, and some examples are given.

In this section, we propose solutions to address each defined network event, aiming to keep the QoS experienced by the user, as well as ensure the SLA definition. However, the SDM-EaaS architecture is flexible to allow the deployment of other solutions.

[1]http://osrg.github.io/ryu/

A. Congestion/Wastage

1) Event Description: The *Congestion/Wastage* network events represent the case in which the resources allocated at the moment are not suitable for the current traffic volume. An example of this case is illustrated in Figure 3, which represents an EaaS environment. The *ISP-EaaS Manager* in Figure 3 shows the entity that manages the EaaS environment, controlling all decisions. In the same way, *T1* and *T2* show the time before and after the adjustment, respectively.

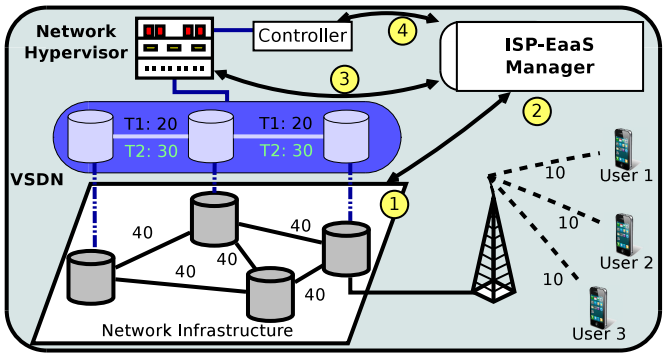

Fig. 3. Example of congestion event.

Figure 3 shows all the steps to perform the management: (1) the congestion is identified, since the traffic demand will be higher than the resource allocation, 30 Mbps (10 Mbps from each user) and 20 Mbps, respectively; (2) the manager consults the policy to be applied when the *Congestion/Wastage* network event occurs; (3) the infrastructure is adjusted according to the policy - in this example, the resource allocation is configured to be the same of the traffic demand; and, (4) the VSDN environment (network hypervisor and the controller) is updated with the new network parameters, in this case the amount of allocated resources.

2) Proposed Approach: This section describes an adjustment policy, which aims to define if it is necessary to perform a resource adjustment based on the traffic demand measurement. Despite the traffic demand information, it uses the M and I values: I is the set of initial resources allocated (defined in the SLA) and M is the upperbound of allowed resource allocation.

The following notation is used to describe the policy behavior: C is the current traffic demand; A is the current resources allocated for the VSDN; F represents the size of the windows to adjust the resource allocation, i.e., amount of resources that will be increased or decreased; T is the percentage of the current amount of resources allocated that is tolerated to be higher (T_H) or lower (T_L) than the current allocation; L is the maximum time that C can stay higher than the current resource allocation, where the network administrator is able to set the value to attend client's particularities. Thus, it considers the environment in one of the following situations according to the value of C:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

- *Congestion*: higher than $\frac{T_H+A}{2}$;
- *Alert*: when $A < C \leqslant \frac{T_H+A}{2}$;
- *Feasible*: when $T_L \leqslant C \leqslant A$;
- *Waste*: lower than T_L.

The defined situations are illustrated in Figure 4. The current traffic demand C can be in one of these situations. In the same way, the current allocated resources (A) is between the initial SLA definition (I) and the maximum available resource in the network infrastructure (M). The definition of M is used to guarantee that the resource allocation will not exceed the available network resource, as well as will not affect the isolation between all the VSDNs, since the resource adjustment will be limited by it.

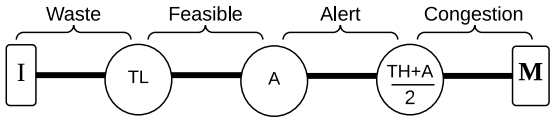

Fig. 4. Possible Situations of the VSDN.

The adjustment policy applies Algorithm 1 to identify and to calculate a possible resource adjustment. It represents the concepts mentioned previously to perform the decision. Hence, it uses the current available infrastructure resources, the situation of the VSDN and the current resource allocation to make the decision. The Adj variable represents the value to be requested from the ISP. When Adj is zero, no request is sent (*Feasible* situation).

Algorithm 1 Adjustment Algorithm

```
1:  Adj = 0;                                   ▷ Adjustment to be requested
2:  if (T_L ⩽ C ⩽ A) then
3:      Feasible situation: adjustment is not necessary.
4:  else
5:      if C > A then
6:          if C < (T_H+A)/2 then
7:              if Long period (L) of Alert Situation then
8:                  Adj = T_H;
9:              end if
10:         else
11:             Congestion situation: Adj = C + F;
12:         end if
13:     else
14:         if C < T_L then
15:             Waste situation: Adj = T_L;
16:         end if
17:     end if
18:     Put Adj under the limit (I ⩽ Adj ⩽ M);
19: end if
```

In general, Algorithm 1 identifies what is the current situation of the resource allocation. It verifies if the current resource allocation is suitable for the current traffic demand (from line 2 to 4), and if it is not, an adjustment is calculated. Line 18 controls if the adjustment made is not lower than the original allocation defined in the SLA (I) or higher than the resource availability at the moment (M).

The adjustment is related to the scenario, where a critical situation (*Congestion*) results in a major change in the resource allocation, according to the factor F. On the other hand, when the traffic demand is not far away from A, a small adjustment is requested. Finally, the algorithm checks if the adaptation is in accordance with the defined acting limits.

SDM-EaaS architecture allows the network administrator to set the frequency of the algorithm execution to determine a possible adjustment. The frequency of execution can affect the scalability of system if the applied policy has a high computational complexity. However, the proposed Algorithm 1 does not compromise the scalability of the architecture, since the complexity of Algorithm 1 is linear.

The periodic information regarding to the current traffic demand (C) is given by the *Passive Monitoring* module, and it is stored under *Infrastructure Monitoring* module. The *Event Handler* checks in which state (Congestion, Alert, Feasible, and Waste) C is according to the *Policy Behavior* module. If any adjustment is necessary, the *Event Handler* informs to the *Central Manager* module, which communicates with *Infrastructure Manager* and *Resource Allocation* modules to update the resource allocation to the VSDN analyzed.

B. Failure

1) Event Description: Users become frustrated when the Internet access fails and the quality level of applications is reduced. Within this context, one important issue to be improved in networking environments is resilience. Resilience has been defined as the capacity of the network to provide a minimum specified level of service in situations of faults in the standard operation [1]. Thus, a *Failure* event represents a situation of fault in standard operation of a network component (link or node). In this case, the VSDNs affected by the failure need to be adjusted to guarantee the QoS for the user and to keep the SLA parameters. Figures 5 and 6 show a situation of failure.

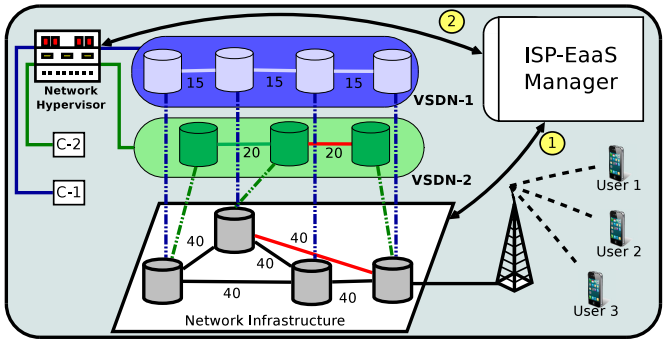

Fig. 5. Failure detection.

Step 1 in Figure 5 illustrates the infrastructure monitoring and the failure identification. After that, the *ISP-EaaS Manager* consults the policy attached to the *Failure* event, and performs the adjustment in the VSDNs affected by the failure (Step 2). In Figure 5, only VSDN-2 needs to be adjusted. Hence, new components are selected to be allocated in the VSDN-2, resulting in the situation illustrated in Figure 6.

2) Proposed Approach: To deploy VSDNs, it is necessary to develop an allocation algorithm that decides which components (links and nodes) will take part on the VSDN to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

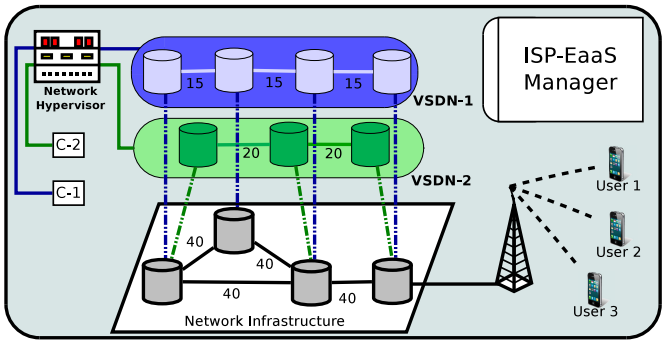

Fig. 6. Environment after failure.

comply with clients' requests. Additionally, the concept of resilience involves not only incorporating reactive actions to manage post-failure consequences, but also pre-event strategic planning.

To deal with failure events and bring resilience to the network, we propose an algorithm to generate one main VSDN topology, plus a backup topology to be deployed when a failure under the main topology happens. The objective of the proposed algorithm is to allocate a VSDN that has the desired bandwidth under normal operating conditions, but is also planned to be resilient under failure events.

In our previous work [20], we developed an algorithm to generate the redundancy inside the VN according to desired reliability. From this existing algorithm, we extended the previous work and developed Algorithm 2 to find one main topology to be used as virtual topology for the VSDN, and one backup topology to be used when necessary. The idea is to define the backup topology periodically, aiming to keep it updated and suitable to the current resource availability in the network.

The following notation is used to model the VSDN allocation problem. Let G be a weighted directed graph representing the network infrastructure. Let D be a set of k designated gateway nodes, and w_l be the cost/weight of network link l. Let $\wp$ represent a quantity close to infinity, and b is the bandwidth requested by the client.

Algorithm 2 Generate VSDN Topology

1: Topology $Main = ShortestPath(G,\ s,\ D, b)$;
2: Copy G to G';
3: **for all** Link $i \in Main$ **do**
4: **for all** Link $j \in G'$ **do**
5: **if** $(i == j)$ **then**
6: $w_j = \wp$;
7: **end if**
8: **end for**
9: **end for**
10: Topology $Backup = ShortestPath(G',\ s,\ D, b)$;

Initially, the $Main$ topology is generated from a shortest path algorithm ($ShortestPath$), i.e., the Dijkstra algorithm defines the paths to reach the destination nodes in D. The w_l of each link in G is 1 if it has at least an amount b of available bandwidth, and the link l is removed from the topology otherwise.

After the definition of the $Main$ topology, a copy of the network infrastructure (G') is created and it is used to update the weight of the links. This update process (from line 3 to 9) aims to avoid the usage of the links already allocated in the $Main$ topology, assigning them with $\wp$ (close to infinity). $\wp$ allows the avoidance of the links in $Main$, but without discarding them as an option, thus forcing the algorithm to seek for alternative paths to reach the desired nodes in the second call of $ShortestPath$ function under the updated topology G' (line 10).

In a nutshell, Algorithm 2 constructs an initial topology, and after that, with the update of link's weight, it seeks alternative paths. The link update is used to avoid allocating edges already being used in the main topology, but still considering them as an option, thus seeking for an alternative paths. If no alternative path exists, the algorithm uses part of the main path due to non existence of a full disjoint path, allowing an allocation that uses the current resource of the network infrastructure.

The proposed approach will interact with the *Passive Monitoring* and the *Infrastructure Monitoring* modules to identify when the network component fails. Therefore, it notifies the *Event Handler* module, which determines the active VSDNs that were affected by the failure. After that, the set of adjustments in the VSDN topologies is informed by the *Central Manager* module, which communicates with *Infrastructure Manager* module to update the VSDN topology allocation in the *Virtual Network Deployment* module. The *Virtual Network Deployment* module is responsible for interacting with the network hypervisor to deploy the new VSDN topology, and *Resource Allocation* updates the resource allocation under the switches in the network infrastructure.

C. Scheduling

1) Event Description: The traffic demand for the edge network follows the social behavior of the users [18]. Thus, the traffic demand changes during the day [23]. Differently from than *Congestion/Wastage* event, the *Scheduling* event represents the situation when an adjustment in the resource allocation is proactively announced or can be accurately predicted. In this way, it is possible to plan this resource allocation, preventing or smoothing the difference between the traffic demand and the current resource allocation for a VSDN. An example of this situation is presented in Figures 7 and 8.

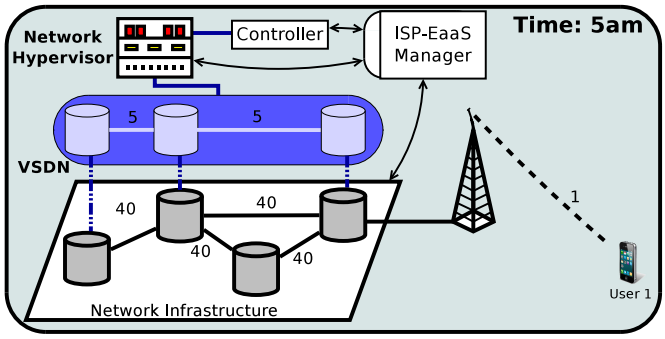

Fig. 7. Low demand period.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

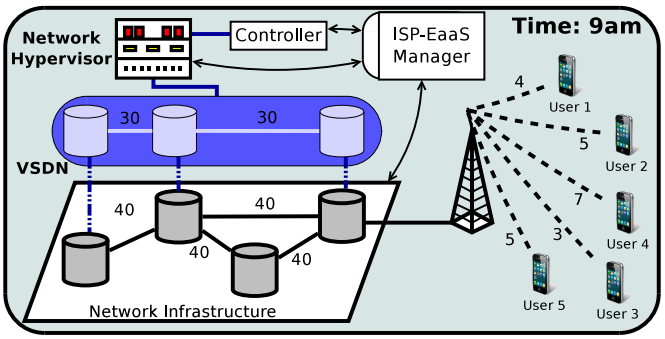

Fig. 8. High demand period.

Figure 7 shows the first hours of the day, where few users generate low traffic. On the other hand, Figure 8 illustrates the time of the day which many users request high amount of resources to the edge network. A good strategy is to plan the resources allocated and the VSDN behavior according to the social behavior, i.e., the traffic demand usually observed during an specific time of the day. The *Scheduling* event represents this idea: the change of the VSDN configuration according to the planned schedule. In this example, the schedule defines that the resources allocation is increased during the period of higher traffic demand.

2) Proposed Approach: As described in Section IV-B2, an allocation algorithm is used to deploy a VSDN and it is necessary to plan the VSDN allocation to improve the resource utilization. Regarding the *Schedule* event, the allocation algorithm must consider the concept of allocation slot. Time slot means that the client does not define a fixed bandwidth to be allocated, but he/she defines a set of configurations that defines the bandwidth to be allocated in a specific slot of time.

To represent the allocation slot, S is the set of allocation slots, where s_i is the bandwidth requested to the time i. In this article, we consider an allocation slot based on hours per day, i.e., 24 slots per day (ranging from 0 to 23). For example, in Figures 7 and 8 the configuration should be $s_5 = 5$ and $s_9 = 30$, respectively. In the same way, let P be the set of topologies, and p_i be the topology allocated for the time i.

To deal with schedule events, we propose Algorithm 3 to generate a VSDN topology for an initial allocation slot, and after that, to try to allocate the VSDN with lower number of changes in the current VSDN in comparison with the previous allocation slot. The idea is to keep as maximum as possible the initial topology with the goal of decreasing the impact of virtual components and flows migration under service provision, since it generates a gap in the service provision during the update process.

We consider that an initial allocation slot will be always provided (i.e., the allocation slot S_0 is always given by the client). This allows the algorithm to have an initial configuration to perform the VSDN allocation. The notation previously defined in Section IV-B2 is applied. Additionally, let ε be a quantity close to zero and b_l be the available bandwidth in link l.

First, the definition of the initial topology P_0 is made according to $ShortestPath$ (described in Section IV-B2), and it is used as the base for the schedule process. P_p is defined

Algorithm 3 Generate the Schedule for a VSDN Topology

```
1:  Topology p_0 = ShortestPath(G, s, D, S_0);
2:  Copy p_0 to p_p;
3:  for all Time Slot i ∈ S do
4:      for all Link l ∈ p_p do
5:          if (b_l ⩽ s_i) then
6:              w_l = ε;
7:          else
8:              w_l = ℘;
9:          end if
10:     end for
11:     Topology p_i = ShortestPath(G, s, D, s_i);
12:     Copy p_i to p_p;
13: end for
```

in line 2, which is implemented to keep the topology of the previous allocation slot.

Algorithm 3 travels through all the allocation slots given by the client, aiming to identify if the previous topology (P_p) fits well to the current allocation slot analyzed (S_i). To check it, from line 4 to 10, Algorithm 3 verifies if the links allocated in P_p (line 5) have the bandwidth requested in current allocation slot to assign ε, or assign ∞ otherwise.

The usage of ε aims to encourage the utilization of link l, since it is suitable to S_i. On the other hand, if the link l can not be used to allocate S_i, its weight (w_l) is set to $\wp$ to enforce the definition of a new topology (in line 11) to avoid link l without discarding it as an option (in the case where no other link exists).

Basically, Algorithm 3 tries to adjust the previous topology just replacing the links which do not fit the bandwidth by other link(s) that are suitable, keeping as maximum as possible the initial topology. This process maintains the current configuration of the switches and the network hypervisor increasing the performance and the delivery capacity of the network, since it tries to avoid a gap in the service provision.

Algorithm 3 is applied to generate the schedule of a single VSDN. Hence, the order to analyze the VSDNs of all clients is defined according to the network administrator, which can be based on pricing, first-in-first-out, SLA duration, and other.

In general, to define the schedule, the *Event Handler* module gets the SLA definition with the desired allocation slot description (S) and it is applied into the *Policy Behavior* module which contains Algorithm 3 to make the scheduling. When the scheduling for the set S is done, this information is stored in *Event Handler*, which controls when a new allocation slot is reached and triggers the adjustment to be made by the *Virtual Network Deployment* and the *Resource Allocation* modules, as in the *Failure* event.

V. Experiments

To evaluate the SDM-EaaS architecture, a prototype was developed. The experiments aim to evaluate the ability of the proposed architecture to address the occurrence of events in the network, including the identification of the event and the application of the proposed method to each specific network event. SDM-EaaS is expected to prevent the decrease in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

quality of experience and to improve resource utilization. The experiments for *Congestion/Wastage*, *Failure*, and *Scheduling* events are described in Sections V-A, V-B, and V-C, respectively.

A. Congestion/Wastage Testbed

The experiments for the *Congestion/Wastage* events were run in a real testbed. The experiments consisted of a set of UDP flows injected from the user to a destination server, in which the VSDN is the interconnection between the user and the server. Figure 9 illustrates the environment configured to perform the experiment. In this experiment the network resource adjustment mechanism described in Section IV-A2 is applied as a *Policy Behavior*.

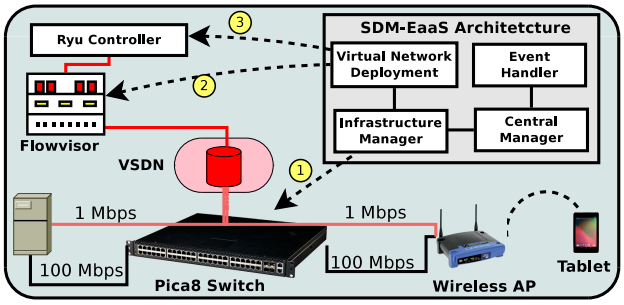

Fig. 9. Testbed configuration

The testbed consisted of a Pica8[2] switch as the SDN infrastructure, a Linksys WRT54GL[3] as client Access Point and a Nexus tablet[4] as the user. The Ryu was the Open Flow controller of the VSDN, while Flowvisor was the network hypervisor. The testbed experiments were performed at the Network Research Laboratory (NRL) of the University of California Los Angeles (UCLA).

Usually, network traffic models assume that the inter-arrival time and duration of flows, which results in traffic demand, follow an exponential distribution [24]. Hence, both the requested bandwidth and duration of requests are generated by an exponential distribution, since both parameters can be considered related to the traffic demand. In this experiment, the following mean values were used: transmission rate = 200Kbps, packet size = 500B , interval between flows = 200ms and duration per flow = 30 seconds. Each experiment generated flows during 60 seconds, i.e., a new flow was started until the 60th second. The network tool Iperf[5] was used to generate the data flows.

The adjustment was applied to the bandwidth of the path connecting the user to the server via the Pica 8 switch. The range is from 1Mbps as minimum (T_L), to 10Mbps as maximum (T_H), 10% of tolerance (T) and 1Mbps as adjustment factor (F). SDM-EaaS was compared to the current static resource allocation approaches (i.e., the bandwidth assigned to each VSDN is the same during the experiment), namely: $1 - Fixed$, $5 - Fixed$ and $10 - Fixed$, with 1Mbps, 5Mbps and 10Mbps, respectively. This static allocation is used as a comparison case because it is the usual approach applied by the ISPs in the service delivery. The experiments were performed 50 times for each case, and the results are presented with a 95% confidence interval.

The difference between the current traffic demand and the resources allocated can generate two situations: (i) wastage, when the client is paying for unused resources; and (ii) degradation of QoS, when the allocated resources are not enough to support the traffic demand, generating packet losses and reducing the QoE of users.

Figure 10 presents the waste of resources in the experiment. The positive values represent wastage situations, while the negative values show the degradation of QoS situations (congestion). The adjustment mechanism enabled by the SDM-EaaS keeps the waste of resources low, avoiding both extremes.

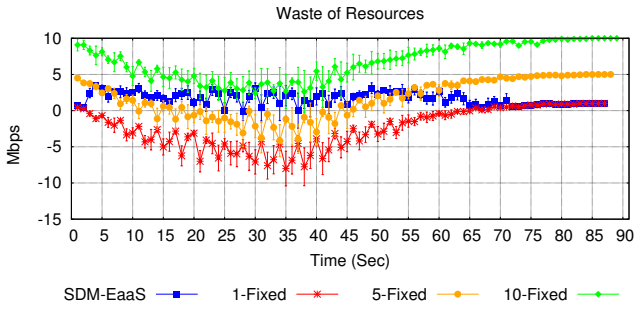

Fig. 10. Difference between allocation and demand

Figure 11 shows the average percentage of loss of all flows that started at each second in the experiment. For example, the loss at the 10th second refers to loss of the flows started at time 10s regardless their configuration. Thus, Figure 11 illustrates the QoS degradation caused by the difference between the traffic demand and the resource allocation.

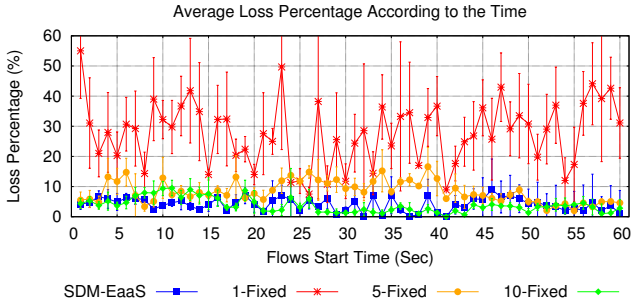

Fig. 11. Average loss

According to the information on the loss, the adjustment mechanism (based on Algorithm 1), the 5Mbps and the 10Mbps cases present low loss percentage. This occurs because 5Mbps and 10Mbps have a static allocation higher than the traffic demand, while the 1Mbps case experienced higher losses. The loss generated by the SDM-EaaS architecture is 5 times lower than the 1-Fixed case, and it is similar to the 5-Fixed and 10-Fixed cases.

In general, the usage of the adjustment process of the SDM-EaaS architecture results in a suitability for the traffic volume

[2]http://www.pica8.com/
[3]http://www.linksys.com/en-eu/products/routers/WRT54GL
[4]http://www.google.com.br/nexus/7/
[5]http://sourceforge.net/projects/iperf/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

and a lower level of loss experienced by the user, while it improves the resource utilization of the SDN infrastructure.

Additionally, we analyzed the behavior of the proposed solution when several VSDNs are running over the network infrastructure. The experiments consisted of 1, 5, 10, 15 and 20 virtual networks running in parallel and a set of traffic is injected in each virtual network, demanding the SDM-EaaS architecture to adjust the VSDNs according to the traffic volume of each one, considering the available bandwidth.

We compared the SDM-EaaS architecture against two fixed approaches: (a) 5 Mbps fixed allocated to each VSDN, independently of the number of actives VSDNs; and (b) a split approach, which splits the bandwidth (100 Mbps in the experiments) between the active networks, for example, when 5 VSDNs are active, each one has 20 Mbps allocated.

Figure 12 shows the average waste of resources in each case. We can observe that the $Split$ approach generates a very high waste when few VSDNs are active. On the other hand, the SDM-EaaS architecture keeps the usage of the resources effective in all cases, representing the capacity of the SDM-EaaS architecture to deal with several VSDNs simultaneously. It is worthy mentioning that in the $20-Case$ the flows injected in each VSDN results in a traffic volume higher than the total available bandwidth, i.e., it is higher than 100 Mbps. This configuration leads the network infrastructure to be saturated, passing by all the possible situations of the resource allocation in front of the traffic volume: waste, feasible and congestion.

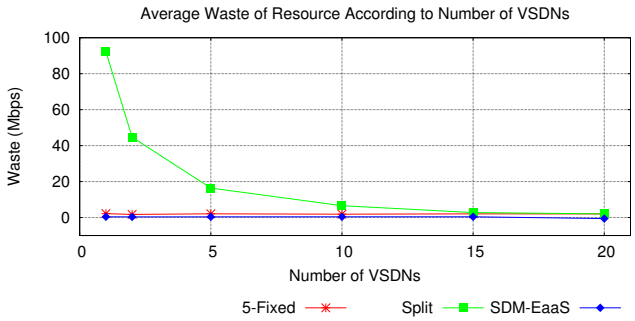

Fig. 12. Average waste of resource

As a consequence of the resource usage, the injected flows suffer several losses, that are presented in Figure 13. In the $5-Fixed$ case, the losses occur more frequent, since during all the experiment the bandwidth allocated is not suitable for the traffic volume.

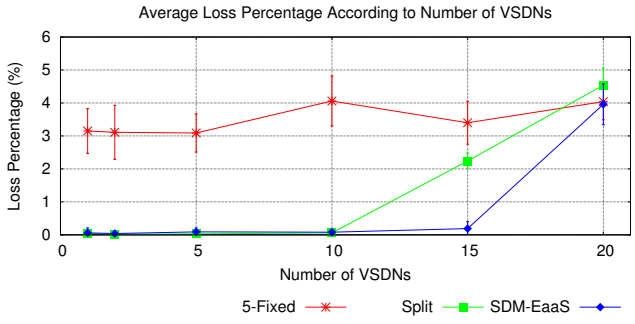

Fig. 13. Average loss

Both $Split$ and SDM-EaaS architecture achieve very low loss percentage in the cases where the few VSDNs are active, but the waste of resource of the $Split$ approach is very high. This fact does not occur in the SDM-EaaS architecture, which keeps the loss percentage low while is possible, since when the 20 VSDNs are active, the traffic volume is higher than the bandwidth of the network infrastructure.

In general, the SDM-EaaS architecture is able to manage the resource allocation to the set of active VSDNs while there is available bandwidth to be used, representing a very low waste of resource. In the same way, it generates benefits to the user's flows that traverse the VSDNs, maintaining the loss percentage low.

B. Experiments of Failure

To evaluate the SDM-EaaS architecture and the proposed solution for *Failure* event, we performed two sets of experiments: (i) an individual evaluation of the VSDN behavior when a failure occurs; and, (ii) the evaluation of connectivity of several VSDNs when failures in the network increase.

1) Individual analysis: The first experiment deployed the scenario presented in Figure 5 in the Mininet[6] emulator, which creates an SDN/Openflow network of virtual hosts, switches, and links. The Mininet was used because it is necessary to have a higher number of links and nodes to deploy the VSDN with both a main and a backup path, which is not possible using a single Pica8 SDN switch, as in the *Congestion/Wastage* experiment.

This experiment aims to evaluate the proposed algorithm in the SDM-EaaS to deal with *Failure* events (described in Section IV-B2) against the no usage of it, identifying the benefits and the overhead under the QoS experienced by the user. Therefore, we injected a TCP flow (using the Iperf tool) of 60 seconds of duration in a virtual network with 100Mbps of resource allocation.

Two situations were emulated to vary the scenario configuration: (i) one link from the main path is configured to shutdown after 30 seconds, and the backup path is deployed; and (ii) two links are configured to shutdown after 30 seconds, one from the main path and another one from the backup path, and the SDM-EaaS has to seek for another alternative path. These configurations aim to evaluate the proposed solution in single failure and multiple failures.

Figure 14 illustrates the data transfer of the TCP flows over the time. When the transmission reaches 30 seconds the link(s) is(are) shutdown, which results in a break in the data transfer. However, when just one failure happens the SDM-EaaS architecture (using Algorithm 2) has a short period of disconnection (around two seconds), which is the time to identify the failure plus the time for the architecture to communicate with the *Network Hypervisor* to update the VSDN components and to perform the resource allocation for the VSDN in the network infrastructure.

On the other hand, when the Algorithm 2 is not used, regardless the number of failures, the communication is disrupted for a long period (around 5 seconds), since it is necessary, besides the steps mentioned previously (to identify the failure and

[6]http://mininet.org/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

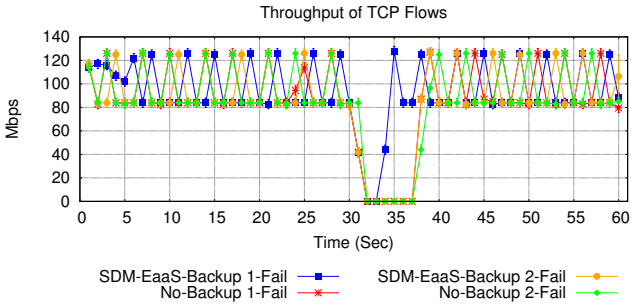

Fig. 14. TCP throughput over time

deploy the new VSDN components), to collect the information about the network infrastructure and calculate the new VSDN topology. When multiple failures compromise the main and backup paths, even with Algorithm 2, it is necessary to perform a new search for the VSDN, resulting in a similar behavior of no backup path definition.

The difference of time to restore the service delivery tends to be higher between the usage of the proposed solution or not according to the number of nodes and links, since the time to search for a new path after failure is directly proportional to the size of the network infrastructure. Usually, the running time of the search task is $O(n^2 + m)$ for n nodes and m links.

Algorithm 2 improves the resilience of the network, allowing a faster re-establishment of the service without overhead for the network, while getting a similar performance when the worst case happens (multiple failures that compromise both main and backup paths). To deal with the worst case, one possible approach is to define more than one backup path, which increases the resilience of the network. However, this approach increases the search time, as well as when the infrastructure topology does not have many network components, it is not possible to find many alternative paths.

2) Connectivity analysis: After the individual evaluation performed in the first experiment, we analyzed the behavior of our solution when several VSDNs are deployed and some network components fail. For this second experiment, we developed a VSDN allocation simulator[7], which is able to perform the following tasks: (i) load an infrastructure topology, in this article we used the Internet2[8] as topology; (ii) manage the infrastructure, as well as resource availability; (iii) define the network for a set of requests according to the allocation algorithm chosen; (iv) generate a set of random failures of network components; and (v) verify the network connection status based on a sequence of failures of network components. Thus, the developed simulator can be used to evaluate the connectivity of VSDN in front of failure events.

The experiments aim to evaluate the capacity of the proposal to allocate a set of one hundred VSDN requests and to analyze the connectivity when failure events occur. Therefore, we evaluate the scenarios according to failure percentage. For example, "5%" means that 5% of components randomly failed.

[7]http://bitbucket.org/rafaellgom/vn-allocation/
[8]http://www.internet2.edu/

The result regarding the number of solved requests that keep full connectivity (reach all desired nodes) after some network component failures is depicted in Figure 15. We observe that the SDM-EaaS keeps a higher connectivity when compared to the traditional single path approach. In all cases, the connectivity status of the set of VSDNs generated by the SDM-EaaS is around 25% higher than the existing approach.

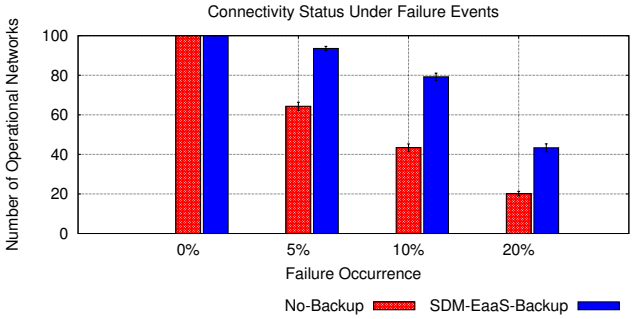

Fig. 15. Connectivity status

In general, the connectivity decreases according to the increment of the number of components that fail. An important point to verify in this analysis is the capacity of our solution to define a set of VSDNs that could keep the connectivity high even when the failure occurrence increases.

C. Experiments of Schedule

The experiment of *Schedule* used the Mininet emulator, where the scenario presented in Figure 7 was deployed. In the experiments, a set of random UDP flows were injected in the network during 30 seconds, and the following mean values were used: 300Kbps of transmission rate, 100ms of interval between flows, and 30 seconds of duration per flow. Additionally, a set of allocation slots are given and compared against a fixed allocation approach (10 Mbps, 25 Mbps, and 50 Mbps).

Although we described Algorithm 3 to work based on hours, during the experiments the algorithm was adapted to work for allocation slots according to seconds, aiming to scale for the period used in the experiments. Following the notation presented in Section IV-C2, the set of allocation slots was: $S_0 = 10$, $S_{10} = 30$, $S_{15} = 40$, $S_{45} = 30$, and $S_{60} = 10$, i.e., initially allocates 10 Mbps, when 10 seconds are reached the allocation is updated to 30Mbps, and so on. The information regarding the resource allocation and the traffic volume generated are shown in Figure 16.

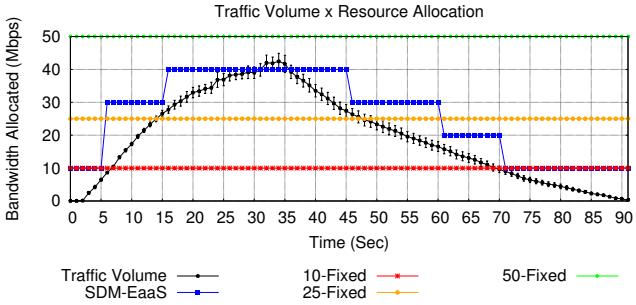

Fig. 16. Resource Allocation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2016.2538821, IEEE Transactions on Network and Service Management

The average percentage of loss of all flows that started at each second in the experiment, as described previously, is presented in Figure 17. As a consequence of the schedule allocation, the loss was very close to zero, similarly to the fixed allocation of 50Mbps.

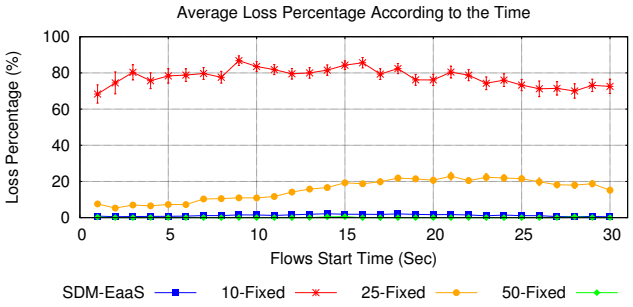

Fig. 17. Average Loss

In general, the planning resulting from the schedule process provides a better utilization of the network resources. However, the performance of the scheduling process is related to the given set of allocation slots. Thus, mis-formulated allocation slots can generate congestion (decreasing the QoS experienced by the user), or can result in wastage of network resources. The definition of the allocation slots can be made according to social behavior [18], previous observation, and others.

D. Final Remarks

The SDM-EaaS architecture was designed to give more flexibility to EaaS environments, mainly due to the capacity to identify and to adapt the VSDN characteristics according to network events, which could decrease the performance of the network. Moreover, the approaches proposed to deal with the events described before can be merged, resulting in a more robust management, as well as extended to incorporate more functionalities.

For example, the schedule approach can define a VSDN with both main and backup paths for each allocation slot of a VSDN, thus merging the solution for the *Schedule* and *Failure* events. Additionally, when a VSDN is operational, an unexpected increase or decrease on the traffic volume could occur, making the current allocation slot not suitable to it. Hence, the *Congest/Wastage* event is triggered and Algorithm 1 could run according to the available resources in the network infrastructure.

It is possible to extend the focus of the solution for each network event according to the network administrator willingness. For example, if the EaaS is designed to interconnect multimedia application, Algorithm 1 could consider an adjustment to be made based on the bitrate of the video flows. Furthermore, if the EaaS receives a set of VSDNs with similar allocation slots, which does not guarantee that all VSDN could have the desired allocation slot in the peak period, two approaches could be deployed: (i) the network administrator first analyzes the high priority VSDNs based on a policy (for example based on higher pricing, or contract duration, etc); and, (ii) the *Congestion/Wastage* event is considered to release network resources temporarily, when wastage is identified.

Using the event concept allows the management of the EaaS environment, focusing directly on the situation that is identified, yet keeping the isolation between the VSDNs. Thus, the network is capable to simultaneously adjust and plan the VSDN individually, but also considering the status of the network infrastructure as a whole. This approach results in a better resource utilization and QoS experienced by the end-user, as the performed experiments suggested.

VI. Conclusion

This article presented SDM-EaaS to enhance the management capacity of EaaS environments, bringing more flexibility and improving resource utilization. A definition and analyses about the network events were presented, as well as how the SDM-EaaS architecture was designed to deal with each network event that occurs in an EaaS environment.

A prototype of the SDM-EaaS architecture was developed and it was evaluated using two approaches: (i) a real testbed representing an EaaS environment for the *Congestion/Wastage* event; and, (ii) a set of experiments under an emulation environment and a simulator for the the *Failure* and *Schedule* events. The objective of the experiments was to evaluate the feasibility of the SDM-EaaS architecture in a ISP, analyzing the behavior of the proposed methods when the number of active VSDNs increase. Based on the experiments, the proposed SDM-EaaS architecture identifies network events and manages the EaaS environment to deal with these events. Thus, applying the proposed architecture, it is possible to guarantee the QoS level, obey the SLA definition, and avoid unnecessary financial expenses.

As future work, we intend to extend the architecture to interact with existing policy frameworks and to perform a larger set of testbed experiments to encompass all the network events described in this article. Additionally, we plan new more robust solutions to each network event described in this article, since the base architecture for VSDN management is defined, for example adding optimization models.

Acknowledgment

The authors would like to thank FAPESP (Sao Paulo Research Foundation - grant 2012/04945-7), CAPES (grant 12342/13-0), RNP and CNPq for the financial support.

References

[1] J. P. Sterbenz, D. Hutchison, E. K. Ãetinkaya, A. Jabbar, J. P. Rohrer, M. Scholler, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245 – 1265, 2010.

[2] S. Davy, J. Famaey, J. Serrat-Fernandez, J. Gorricho, A. Miron, M. Dramitinos, P. Neves, S. Latre, and E. Goshen, "Challenges to support edge-as-a-service," *Communications Magazine, IEEE*, vol. 52, no. 1, pp. 132–139, January 2014.

[3] S. Yeganeh, A. Tootoonchian, and Y. Ganjali, "On scalability of software-defined networking," *Communications Magazine, IEEE*, vol. 51, no. 2, pp. 136–141, February 2013.

[4] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T.-Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, S. Seetharaman, D. Underhill, T. Yabe, K.-K. Yap, Y. Yiakoumis, H. Zeng, G. Appenzeller, R. Johari, N. McKeown, and G. Parulkar, "Carving research slices out of your production networks with openflow," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 129–130, January 2010.

[5] A. Al-Shabibi, M. De Leenheer, M. Gerola, A. Koshibe, G. Parulkar, E. Salvadori, and B. Snow, "Openvirtex: Make your virtual sdns programmable," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14. New York, NY, USA: ACM, 2014, pp. 25–30.

[6] H. Kim and N. Feamster, "Improving network management with software defined networking," *Communications Magazine, IEEE*, vol. 51, no. 2, pp. 114–119, February 2013.

[7] R. Mijumbi, J. Serrat, J.-L. Gorricho, and R. Boutaba, "A path generation approach to embedding of virtual networks," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 3, pp. 334–348, Sept 2015.

[8] M. Grossglauser and D. N. C. Tse, "A framework for robust measurement-based admission control," *IEEE/ACM Trans. Netw.*, vol. 7, no. 3, pp. 293–309, Jun. 1999.

[9] S. Song, S. Hong, X. Guan, B.-Y. Choi, and C. Choi, "Neod: Network embedded on-line disaster management framework for software defined networking," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, May 2013, pp. 492–498.

[10] C.-C. Lo, H.-H. Chin, M.-F. Horng, Y.-H. Kuo, and J.-P. Hsu, "A flexible network management framework based on openflow technology," in *Modern Advances in Applied Intelligence*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, vol. 8482, pp. 298–307.

[11] D. Tuncer, M. Charalambides, S. Clayman, and G. Pavlou, "Adaptive resource management and control in software defined networks," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 1, pp. 18–33, March 2015.

[12] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "A virtual sdn-enabled lte epc architecture: A case study for s-/p-gateways functions," in *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for*, Nov 2013, pp. 1–7.

[13] T. Lin, J.-M. Kang, H. Bannazadeh, and A. Leon-Garcia, "Enabling sdn applications on software-defined infrastructure," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*, May 2014, pp. 1–7.

[14] H. T. Nguyen, A. V. Vu, D. L. Nguyen, V. H. Nguyen, M. N. Tran, Q. T. Ngo, T.-H. Truong, T. H. Nguyen, and T. Magedanz, "A generalized resource allocation framework in support of multi-layer virtual network embedding based on {SDN}," *Computer Networks*, vol. 92, Part 2, pp. 251 – 269, 2015, software Defined Networks and Virtualization.

[15] J. He, R. Zhang-Shen, Y. Li, C.-Y. Lee, J. Rexford, and M. Chiang, "Davinci: Dynamically adaptive virtual networks for a customized internet," in *Proceedings of the 2008 ACM CoNEXT Conference*, ser. CoNEXT '08. New York, NY, USA: ACM, 2008, pp. 15:1–15:12.

[16] I. Bueno, J. Aznar, E. Escalona, J. Ferrer, and J. Antoni Garcia-Espin, "An opennaas based SDN framework for dynamic qos control," in *IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–7.

[17] H. Li, X. Que, Y. Hu, X. Gong, and W. Wang, "An autonomic management architecture for sdn-based multi-service network," in *Workshops Proceedings of the Global Communications Conference, GLOBECOM 2013, Atlanta, GA, USA, December 9-13, 2013*, 2013, pp. 830–835.

[18] M. Zhang, C. Wu, Y. Qiang, and M. Jiang, "Robust dynamic bandwidth allocation method for virtual networks." in *Proceedings of IEEE International Conference on Communications (ICC)*, 2012.

[19] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira, "A generic sla negotiation protocol for virtualized environments," in *Proceedings of 18th IEEE International Conference On Networks (ICON)*, 2012.

[20] ——, "A virtual network allocation algorithm for reliability negotiation," in *22st International Conference on Computer Communications and Networks (ICCCN)*, 2013.

[21] P. Kazemian, M. Chang, H. Zeng, G. Varghese, N. McKeown, and S. Whyte, "Real time network policy checking using header space analysis," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, ser. nsdi'13. Berkeley, CA, USA: USENIX Association, 2013, pp. 99–112. [Online]. Available: http://dl.acm.org/citation.cfm?id=2482626.2482638

[22] D. Verma, "Simplifying network administration using policy-based management," *Network, IEEE*, vol. 16, no. 2, pp. 20–26, Mar 2002.

[23] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling internet traffic dynamics of cellular devices," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, ser. SIGMETRICS '11. ACM, 2011, pp. 305–316.

[24] T. M. Chen, *Network Traffic Modeling*. John Wiley & Sons, Inc., 2007.

Rafael L. Gomes is a PhD Candidate in Computer Science at the University of Campinas (UNICAMP) at Brazil. He was a research visitor at Network Research Lab from University of California Los Angeles (UCLA) in 2014. He is bachelor in Computer Science, and he has experience and works on the following topics: Wireless Mesh Networks, Traffic Classification, Network Management, Service Level Agreements (SLA), Virtualization, Resilience, Software Defined Networks and Future Internet.

Luiz F. Bittencourt received Bachelor degree in Computer Science from the Departament of Informatics at the Federal University of Parana, and he received his PhD in Computer Science from University of Campinas (UNICAMP). Currently, he is an Assistant Professor at the Institute of Computing - UNICAMP. He has received the IEEE Latin American Young Professional Award 2013. His research interests are on aspects of scheduling in grid/cloud computing systems, underlying network infrastructure, virtualization, future internet, green computing, and access networks.

Edmundo R. M. Madeira is a Full Professor at the University of Campinas (UNICAMP), Brazil. He received his Ph.D. in Electrical Engineering from UNICAMP in 1991. He has published over 150 papers in national and international conferences and journals. He was the General Chair of the 7th Latin American Network Operation and Management Symposium (LANOMS'11), and he was a Technical Program Co-chair of the IEEE LatinCloud'12. He is a member of the editorial board of Journal of Network and Systems Management (JNSM), Springer. His research interests include network management, future Internet and cloud computing.

Eduardo Cerqueira received Master in Computer Science at Federal University of Santa Catarina, Brazil (2003) and his PhD in Informatics Engineering from the University of Coimbra, Portugal (2008). He was an invited auxiliary professor at the Department of Informatics Engineering of the University of Coimbra (2008-2009). He is an associate professor at the Faculty of Computer Engineering of the UFPA in Brazil and now researcher at Network Research Las at UCLA/USA and Centre for Informatics and Systems of the University of Coimbra (CISUC)/Portugal. He is involved in the organization of several international conferences and workshops, including Future Multimedia Networking (IEEE FMN), Future Human-centric Multimedia Networking (ACM FhMN), ICST Conference on Communications Infrastructure, Systems and Applications in Europe (EuropeComm), Latin America Conference on Communications (IEEE LATINCOM) and Latin American Conference on Networking (IFIP/ACM LANC). He has been serving as a Guest Editor for 5 special issues of various peer-reviewed scholarly journals. His research involves Multimedia, Future Internet, Quality of Experience, Mobility and Ubiquitous Computing.

Mario Gerla received a graduate degree in engineering from the Politecnico di Milano, in 1966, and the M.S. and Ph.D. degrees in engineering from UCLA in 1970 and 1973, respectively. From 1973 to 1976, Dr. Gerla was a manager in Network Analysis Corporation, Glen Cove, NY, where he was involved in several computer network design projects for both government and industry, including performance analysis and topological updating of the ARPANET under a contract from DoD. From 1976 to 1977, he was with Tran Telecommunication, Los Angeles, CA, where he participated in the development of an integrated packet and circuit network. Since 1977, he has been on the Faculty of the Computer Science Department of UCLA. His research interests include the design, performance evaluation, and control of distributed computer communication systems and networks. His current research projects cover the following areas: design and performance evaluation of protocols and control schemes for Ad Hoc wireless networks; routing, congestion control and bandwidth allocation in wide area networks, and; traffic measurements and characterization.