Enabling highly dynamic mobile scenarios with Software Defined Networking

Alberto Huertas Celdrán, Manuel Gil Pérez, Félix J. García Clemente, and
Gregorio Martínez Pérez, *Member, IEEE*

***Abstract*—Mobile devices have promoted the users' mobility and, therefore, the necessity of providing services that accomplish the users' requirements at any place and time. With this, location becomes a key aspect to provide the dynamism required by solutions like the provisioning of reasonable mobile services by service provider networks. In that sense, the Software Defined Networking (SDN) paradigm arose to evolve from current static networks, where they are manually configured by administrators, towards dynamic networks able to manage by their own at run-time and on demand. Solutions managing the SDN resources by using policies have been proposed, but they do not consider one of the main aspects to network dynamism, i.e. the mobility. This article presents a mobility-aware and policy-based on demand control network solution oriented to the SDN paradigm. This is in charge of managing at run-time the service and/or system state with high-level policies, which consider the mobility of users and services, the network statistics, and the infrastructure location. In this context, we define different use cases with the concerns that end-users find when they are in very crowded places, and the solutions provided by our solution through policies: balancing the network traffic between the infrastructure located close to the overloaded one; creating or dismantling geolocated virtual network infrastructure when the existing one is not enough, or is misused to accomplish the end-user demand; and restricting specific network traffic in critical scenarios, like in sport events where crowd consume services with a large bandwidth.**

***Index Terms*—Software Defined Networking, dynamic scenarios, mobility, management-oriented policies.**

I. Introduction

The recent technology advancements in mobile devices and networks have encouraged users' mobility, thus being location one of the most important aspects for knowing where devices, resources, or people are. Location information can provide useful evidence with which to develop new proposals and solutions. For example, the European Commission is making great efforts, funding the Horizon 2020 Programme to define new use cases where mobility and dynamism are key aspects. Under the 5G-PPP initiative, the EU project METIS-II (*Mobile and wireless communications Enablers for the Twenty-twenty Information Society*) [1] is proposing several use cases that highlight the provisioning of reasonable mobile broadband by service provider networks, with high levels of service experience in crowded areas (e.g., stadiums or shopping malls) and even with end-users on the move (e.g., in cars or trains). Other initiatives are being conducted in parallel in other countries or continents, such as 4G Americas [2], where leading telecommunications service providers and manufacturers are fostering the advancement of the LTE mobile broadband technology and its evolution beyond to 5G, or the IMT-2020 (5G) Promotion Group [3] including main operators, vendors, research institutes in China.

Managing the dynamism displayed by the previous proposals requires a deep change from the current networks, where service provider administrators usually configure the network depending on triggered events, towards Self-Organized Networks (SON) [4], which are able to monitor, manage, and configure by their own at run-time and depending on different factors, among which location of users receiving a service is a critical one. This diversity requires that service provider networks collect and analyze large quantities of data, thereby increasing the network management complexity.

In order to ease the network management arose the Software Defined Networking (SDN) paradigm [5]. SDN is a paradigm where a central software program, called *controller*, is the brain of the network to manage its behavior, thereby making network devices become simple packet forwarding elements. This paradigm focuses on the separation of the *control plane* (where the controller is) from the *data plane* (where the forwarding devices are); the definition of a logically centralized controller; the use of open interfaces between the control and data planes; and the programmability of the network by applications. These features provide several benefits, such as the ease to change the network configuration through software rather than typing commands in network devices. Nowadays, we can find several solutions focused on deciding how the SDN resources have to be managed at run-time.

For example, NetGraph [6] provides a scalable graph library and its interfaces with the controller to support network management functions, such as run-time monitoring and diagnostics. Another example is Procera [7], an event-driven network control framework that uses high-level policies to manage and configure the network state. This solution enables dynamic policies, which are translated into a set of forwarding rules to manage the network state by the controller. Following

This work has been supported by a Séneca Foundation grant within the Human Resources Researching Training Program 2014, the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672 - SELFNET (*Framework for Self-Organized Network Management in Virtualized and Software Defined Networks*), the Spanish MICINN (project DHARMA, *Dynamic Heterogeneous Threats Risk Management and Assessment*, with code TIN2014-59023-C2-1-R), and the European Commission (FEDER/ERDF).

A. Huertas Celdrán, M. Gil Pérez, and G. Martínez Pérez are with the Departamento de Ingeniería de la Información y las Comunicaciones, University of Murcia, 30071 Murcia, Spain (e-mail: alberto.huertas@um.es; mgilperez@um.es; gregorio@um.es)

F. J. García Clemente is with the Departamento de Ingeniería y Tecnología de Computadores, University of Murcia, 30071 Murcia, Spain (e-mail: fgarcia@um.es)

the policy-oriented approach, we find OpenSec [8]. Opensec is an OpenFlow-based framework that allows the network operators to describe security policies using human-readable language to implement them across the network.

Up until this point, we have seen that there are solutions allowing the SDN controller to manage the network resources at run-time, using policies defined by service provider network administrators beforehand. Yet, these solutions do not consider one of the main aspects that provide network dynamism, i.e. the mobility. We think that it is a must to consider users' mobility and the location of the network resources so as to manage and configure the SDN state in a more accurate way. In that sense, this paper presents a mobility-aware and policy-based on demand control network solution oriented to the SDN paradigm. Specifically, our solution is in charge of managing the SDN resources at run-time, using high-level policies that consider the mobility of users and services, the network statistics, and the infrastructure location. These policies are oriented to guarantee end-users experience in very crowded places (e.g., stadiums, shopping malls, or unexpected traffic jams). To this end, the policies decide when the SDN should balance the network traffic between the infrastructure located close to the congested one; when the SDN should create or dismantle physical or virtual infrastructure in case of the congested one is not enough to accomplish the end-user demand; and when the SDN should restrict or limit specific services or network traffic in critical situations produced by large crowds using services at specific areas.

II. Use cases in a dynamic mobile scenario

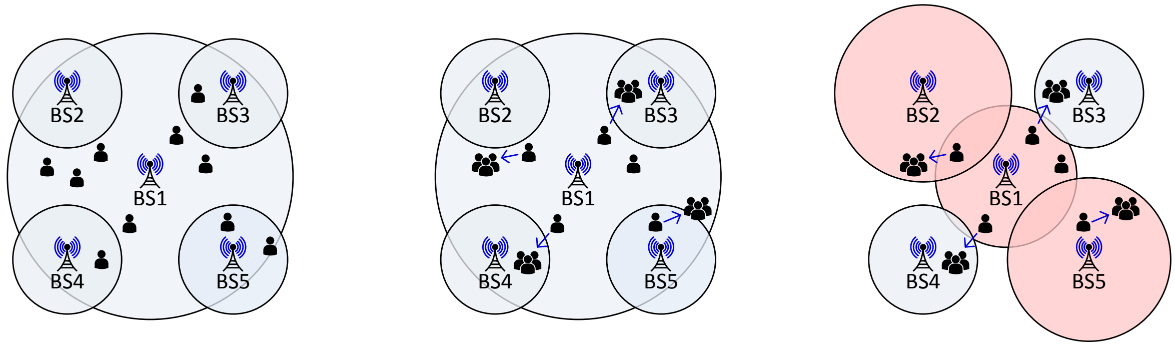
This section shows a dynamic mobile scenario composed of four different use cases, with which to illustrate the service provisioning concerns that end-users can find when they are in a very crowded place (e.g., open air festivals, traffic jams, stadiums, or public events with lots of people). The first use case shows a concern when the network provides low quality services, even having enough resources to accomplish the end-users requirements. The second use case considers that the network does not have enough resources and provides low quality services, whereas in the third use case the network does not have enough resources and it is not able to provide services. In the fourth use case, the network misuses its resources to provide services. In Section III-B, we will explain in detail how our solution manages these concerns to ensure end-users experience.

A use case showing the first concern is shown in Fig. 1a, where a central base station (BS1) and four secondaries (BS2, BS3, BS4, and BS5) are located along a specific area. When large crowds are formed, and end-users move across the networking area, the BS1 is overloaded. Fig. 1b shows this situation. BS1 is congested because it is providing services to a lot of users, and BS2 and BS5 just to a few. To solve it, our solution allows the load balancing at run-time between the base stations located close to the congested one (BS1). In that sense, Fig. 1c shows how the zoom cell size load balancing technique [9] decreases the BS1 cell size and increases BS2 and BS5 cell sizes to ensure end-users experience. It is worth noting that when the crowd moves inside or outside the area, our system dynamically balances the load traffic increasing or decreasing the size of the base stations cells. An example of this situation could be an open festival with a central base station covering the whole festival, and four base stations close to the concert stages. Once the concerts start, the crowd moves to the concert stages and overload the central base station (e.g., sharing photos and videos through social networks).

Regarding the second concern, produced when the network does not have enough resources and provides low quality services, Fig. 2a shows a use case where a base station (BS1) and four generic hardware elements (HW) with 3G/4G antennas are located along a specific area. In this context, Fig. 2b shows the moment when a mobile crowd is formed and the BS1 cannot accomplish the end-users requirements. To manage this situation, our proposal allows creating virtual base stations (BS2, BS3, BS4, and BS5) at run-time by using at will the generic hardware elements. Fig. 2c, depicts the situation managed by our solution. The created virtual base stations are providing services once the network traffic is balanced. It is worthy to note that once the crowd is gone our proposal dismantles the virtual base stations, and the generic hardware will be available to the service provider network. This situation is shown in Fig. 4, which is explained in detail at the end of this section. An example of this second use case, could be a motorway with a base station and four generic hardware elements located along its area. Due to weather conditions, a traffic jam is formed and the base station cannot accomplish the requirements of the crowd, even knowing the atmospheric forecast. To solve it, our solution decides to create four virtual base stations from the existing generic hardware and balances the traffic between them.

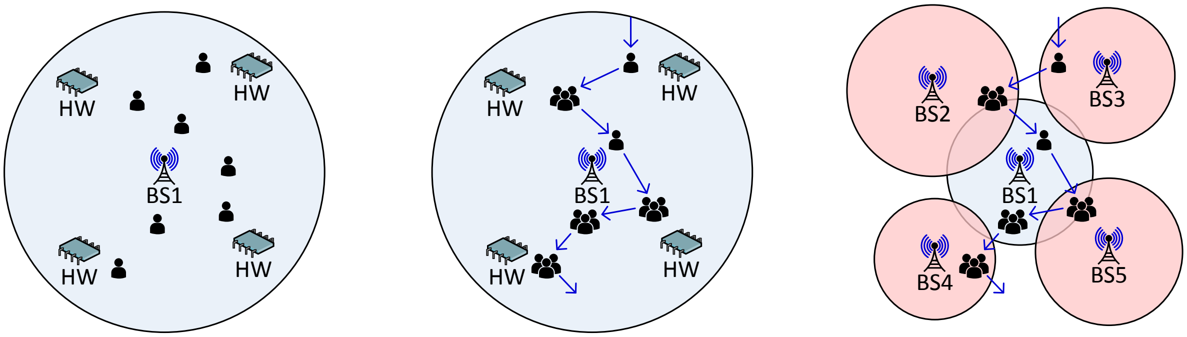
The use cases commented earlier may become critical situations when the network does not have more available resources to accomplish the crowd necessities. In this sense, Fig. 3a shows a new use case where the whole available network infrastructure (all the base stations) is already deployed in a certain area to ensure the end-users experience. Fig. 3b depicts how this situation could become critical causing the network cannot provide services when more users come and consume services that require a large bandwidth like, for example, 4K Ultra High Definition (UHD) video. To solve it, Fig. 3c shows the scenario, where our solution decides that all the base stations reduces the quality of video service from 4K Ultra High Definition (UHD) to High Definition (HD), and limits the bit rate to decrease the network congestion. As in the previous use cases, the reverse process (restrictions are removed) is performed when crowd conditions disappear. An example of this use case could be the Super Bowl event, where the whole network infrastructure is deployed and balanced along the stadium. At the celebration, the crowd massively makes use of the network to send 4K-UHD videos, thus causing the base stations cannot accomplish the demand.

Up until now, we have seen several concerns generated when large crowds are formed. However, it is important to consider the reverse process, when the crowds are gone and the resources are not used in an efficient way, wasting energy resources. In that sense, the fourth concern arisen when



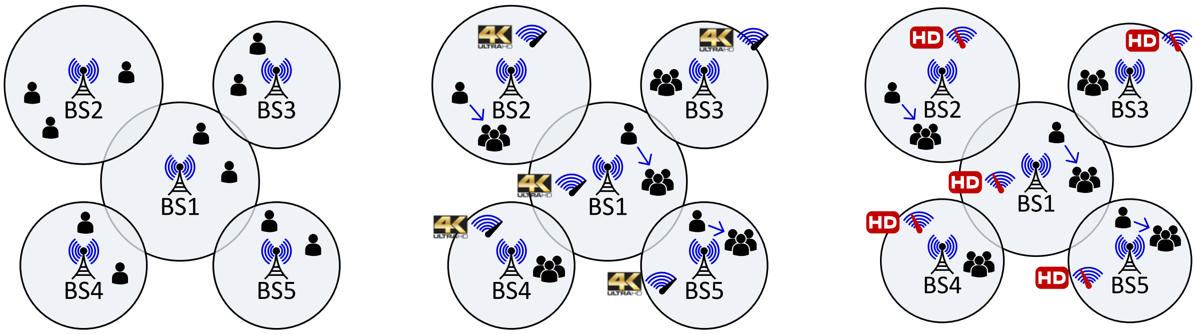
(a) Initial scenario (b) Concerned scenario (c) Managed scenario

Fig. 1. Network with enough resources providing low quality services in a crowded scenario.



(a) Initial scenario (b) Concerned scenario (c) Managed scenario

Fig. 2. Network without enough resources providing low quality services in a crowded scenario.



(a) Initial scenario (b) Concerned scenario (c) Managed scenario

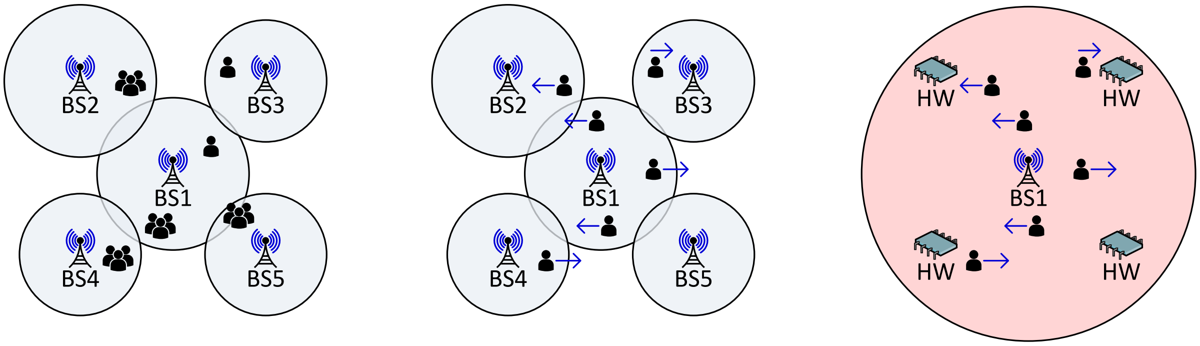
Fig. 3. Network without more resources unable to provide services in a crowded scenario.

the network uses unnecessary resources to provide services. Fig. 4a shows a use case where a physical base station (BS1) and four virtual base stations (from BS2 to BS5) are providing services along a specific crowded area. Fig. 4b shows the moment when the crowd starts leaving the area and all base stations continue providing services to a few users. In order to prevent the misuse of resources, our proposal allows dismantling the virtual base stations (BS2, BS3, BS4, and BS5) at run-time. Fig. 4c depicts this situation. The virtual base stations are dismantled and BS1 provides services after increasing its cell size through a load balancing.

Following with the traffic jam example, the jam begins clearing up when the weather conditions improve, and the virtual network infrastructure previously created is not necessary. In that sense, our solution decides to dismantle the four virtual base stations and balance their traffic to BS1 by increasing its cell size to cover the whole motorway area.

III. SDN MANAGEMENT POLICIES

The policy-based management lets the simplification and automation of the network administration processes [10]. By using policies, the SDN paradigm can control the network state at run-time and on demand in order to guarantee the end-user experience. Among the different sets of policies, we emphasize here the use of mobility-aware management-oriented policies, defined by the service provider network



(a) Initial scenario (b) Concerned scenario (c) Managed scenario

Fig. 4. Network misusing resources to provide services.

TABLE I
ELEMENTS THAT COMPOSE THE BASE OF OUR MOBILITY-AWARE MANAGEMENT POLICIES

Element	Values	Description
Type	Load balancing, Infrastructure, Restriction	Indicate the kind of policy
Resource	Base station, Switch, Service, Intrusion Detection System, etc.	Network element whose information is being managed
Metric	Average of Bytes per flow (ABf), Average Number of Packets per Flow (ANPPF), Average of Duration per flow (ADf), etc.	Define the term that encompasses the different parameters that can be used to evaluate the network state
Location	Geographic position, Area, etc.	Position or region where the policy will be enforced
Date	Date, Hour, Timestamp, etc.	Moment or period of time at which the policy will be applied
Result	Balance, Create, Dismantle, Disable, Limit	Action performed over the network when the policy is applied

administrator to decide the actions made by the SDN according the network infrastructure statistics and location, and the mobility of users and services. In our solution, the schema of the rules shaping the policies are composed of the elements shown in TABLE I, being

$$Type \wedge Resource \wedge Metric \wedge Location \wedge Date \rightarrow Result$$

A. Policies to guarantee end-users experience

We introduce below the three kinds of policies required to manage the concerns depicted in the previous use cases, although other sorts of policies could be defined at will because the proposed solution herein presented is extensible.

1) Load balancing policies: These policies are in charge of deciding when, where, and why it is needed a load balancing of the traffic between the network resources, this being a key aspect in the SDN paradigm for managing and forwarding at run-time the packets passing by the network, considering their location, the date, and the metrics previously defined. These parameters are optional in this kind of policies. It is important to note that we are not proposing a new load balancing solution, but ours is able to use any load balancing solution.

2) Infrastructure policies: These policies allow the SDN paradigm to create or dismantle virtual network resources located at specific locations. As the previous kind of policies, they can be applied in a proactive way in case of knowing when the network needs more infrastructure. As before, the *Date*, *Metric*, and *Location* parameters are also optional.

3) Restriction policies: They manage the network or SDN to guarantee the end-user experience. These policies allow the SDN paradigm to disable or limit the traffic of given network resources or services in case the traffic overload is critic.

B. Managing the dynamic mobile scenario

It is shown below how our solution manages the concerns presented in Section II and how we guarantee end-user experience in very crowded places, when important changes in the population are produced in a short period of time.

Regarding the first concern, when the network has enough resources but it provides low quality services, our solution defines a generic load balancing policy. The policy defined below indicates, for example, that when the ABf value of any base station is within *Yellow* range values (the range of this alarm is set by the service provider administrator depending on the state and characteristics of the scenario), the network should try to balance the traffic load between the base stations located at the same area as the congested ones.

```
Type(#LoadBalancing) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
hasABf(?bs,?abf) ∧ inRange(?abf,#Yellow)
→ balance(?bs,?nearBs)
```

In this policy, *BaseStation* is a possible value of the *Resource* element (defined in our policy schema as shown in TABLE I); *Location* and *locatedBaseStation* are modeled by the *Location* element; *hasABf* is a specific *Metric*; and *balance* is a possible value of the *Result* element. Considering our open air festival scenario, Fig. 1c shows in red the changes made by this policy in the festival area.

To manage the second concern, when the network does not have enough resources and provides low quality services, our solution defines an *Infrastructure* policy. As an example, the policy defined below creates new virtual base stations from generic hardware located close to the congested one when ANPPF of any base station is within *Orange* range values (this alarm is also defined by the service provider administrator, whose range of values is higher than *Yellow* range).

```
Type(#Infrastructure) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedResources(?area,?resource) ∧
hasANPPF(?bs,?anppf) ∧ inRange(?anppf,#Orange)
→ create(?resource,#BaseStation)
```

In this policy, *BaseStation* is a value of the *Resource* element; *Location* and *locatedResources* are shaped by the *Location* element; *hasANPPF* is a kind of *Metric*; and *create* makes reference to a possible value of the *Result* element. Following the traffic jam scenario, Fig. 2c depicts in red the virtual base stations (BS2, BS3, BS4, and BS5) created from the existing generic hardware. Furthermore, it is necessary a new load balancing policy once the virtual base stations are created, in order to balance the network traffic between them.

Regarding the third concern, when the network does not have more resources and it cannot provide services, our solution avoids this situation with two *Restriction* policies. The first one is in charge of disabling the 4K-UHD video traffic of the base stations located at the congested area. It is important to note that a disable action does not filter the video service, but disables a specific quality and the service is provided with lower quality. Below we can find this policy.

```
Type(#Restriction) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
Service(?nearBs,?service) ∧
hasABf(?bs,?abf) ∧ inRange(?abf,#Red)
→ disable(?service,#4K-UHDVideo)
```

The second *Restriction* policy limits the bit rate of the services provided by the base stations located in the congested area.

```
Type(#Restriction) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
Service(?nearBs,?service) ∧
hasABf(?bs,?abf) ∧ inRange(?abf,#Red)
→ limit(?service,#BitRate)
```

In both policies, *BaseStation* and *Service* are values of the *Resource* element; *Location* and *locatedBaseStation* are modeled by the *Location* element; *hasABf* is a kind of *Metric*; and *disable* and *limit* are values of the *Result* element. Fig. 3c depicts the Super Bowl event, where all base stations located at the stadium area decrease the video quality (from 4K-UHD to HD) and limit the bit rate.

Finally, the fourth concern arises when crowd is gone and the network resources are misused. Our solution defines an *Infrastructure* policy that dismantles the misused virtual base stations located close to the underloaded one when the ANPPF value of any base station is less than *Yellow* range values.

```
Type(#Infrastructure) ∧ BaseStation(?bs) ∧
Location(?bs,?area) ∧ locatedBaseStation(?area,?nearBs) ∧
hasANPPF(?bs,?anppf) ∧ lessRange(?anppf,#Yellow) ∧
hasANPPF(?nearBs,?nearAnppf) ∧
lessRange(?nearAnppf,#Yellow)
→ dismantle(?nearBs,#BaseStation)
```

As before, *BaseStation* is a value of the *Resource* element; *Location* and *locatedBaseStation* are shaped by the *Location* element; *hasANPPF* is a kind of *Metric*; and *dismantle* corresponds to a value of the *Result* element. Following the traffic jam scenario, Fig. 4c shows the virtual base stations (BS2, BS3, BS4, and BS5) dismantled and converted again in generic hardware (HW). Furthermore, it is necessary a new load balancing policy once the virtual base stations are dismantled, in order to balance the network traffic to BS1.

IV. Architecture

This section describes our mobility-aware architecture for managing networks oriented to the SDN paradigm at run-time and on demand. Fig. 5 shows the proposed architecture, where the *SDN plane* contains the elements forming the layers of the SDN paradigm and the *SDN management plane* depicts the components composing our solution.

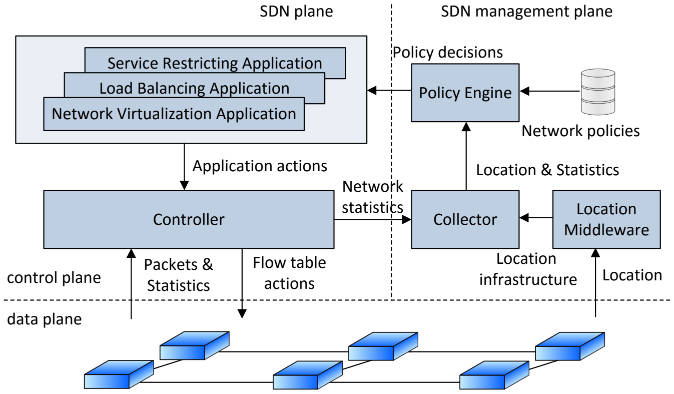


Fig. 5. Architecture of the proposed mobility-aware and policy-based solution

A. SDN plane

One of the main features of SDN is the decoupling of the control from the data plane. In that sense, our proposal has the *data plane* at the bottom layer, where physical and virtual network infrastructure (base stations, switches, routers, etc.) forwards and manipulates packets, not having any control intelligent. The networking logic control is allocated in the *control plane*, in which the Controller component lies on.

To exchange information between control and data planes, our solution makes use of OpenFlow [11]. This is one of the most common southbound SDN interfaces and allows our

Controller to get statistical data about the network traffic, as well as the management of the network infrastructure through software. Nowadays, there are a high number of OpenFlow-capable controllers, such as OpenDaylight, which is used by our solution.

Finally, the application layer is at the top of the SDN stack. This layer contains the applications that use the services provided by the Controller to perform tasks related to the network. Among the existing applications, we highlight three of them used in our solution. The Network Virtualization Application is in charge of managing the virtual network resources by using a well-known open-source software platform called OpenStack Networking (Neutron). Other solutions can be found in the literature such as FlowN [12], which presents an architecture for SDN virtualization. This allows tenants to specify their own address space, topology, and control logic. The second application is the Load Balancing Application, which redistributes the network traffic between the network resources. In this topic, several solutions have been proposed, as the one presented in [9], where load balancing is performed increasing or decreasing the cell size according to the traffic load, user requirements, and network conditions. The last application is the Service Restriction Application, which restricts the network traffic by considering different parameters, such as the bit rate, services, ports, etc.

B. SDN management plane

The main component of our solution is the Policy Engine. This component is in charge of making decisions over the SDN applications, considering network statistics, the infrastructure location information, and the network policies. Among the possible decisions, we highlight three of them. The first one consists on notifying the Load Balancing Application about the need of redirecting the traffic. The second one is focused on deciding if Network Virtualization Application has to create or dismantle virtual resources. The last decision is aimed at knowing if the Service Restriction Application should limit or disable some kind of traffic.

To perform the previous decisions, the Policy Engine uses network policies, defined by the service provider network administrator, and geospatial network statistic information provided by the Collector. This component generates geospatial network statistics, by joining the information received from the Controller and the infrastructure location obtained from the Location Middleware. In order to deploy the Collector, we have several options like, for example, the extended version of IPFIX that includes the location of the network infrastructure to generate network statistics.

Finally, the Location Middleware component obtains the locations of the network infrastructure. This is an independent middleware that provides independence to our solution with regard to the location system used, thus allowing the Location Middleware to choose the best location system or middleware depending on the environment.

V. Conclusion and future work

This paper has presented a mobility-aware solution to manage at run-time networks oriented to the SDN paradigm, considering users' mobility as a key aspect for the service provision. This proposal uses management policies to decide on demand the actions performed by the network, considering the mobility of users and services, the network statistics, and the infrastructure location. These policies ensure the end-user experience in crowded scenarios balancing the network traffic between the infrastructure located close to the congested one, when the SDN has enough resources but it provides low quality services; creating virtual network infrastructure when the SDN does not have enough resources and provides low quality services; and restricting specific network traffic, when the SDN does not have more resources and it is unable to provide services.

As next steps of research, we plan to validate our solution in a 5G advanced self-organizing network, as this has an important intelligence component oriented to the SDN paradigm. This scenario is proposed in the EU project for 5G called Selfnet, which is included in the 5G-PPP initiative and where the authors of this paper are currently working.

References

[1] European Commission, "The EU project METIS-II," [Online]. Available: https://metis-ii.5g-ppp.eu.

[2] 4G Americas, "The voice of 5G for the Americas," [Online]. Available: http://www.4gamericas.org.

[3] "The Chinese IMT-2020 (5G) Promotion Group," [Online]. Available: http://www.imt-2020.cn/en.

[4] H. Yang, X. Meng, and S. Lu, "Self-organized network-layer security in mobile ad hoc networks," in Proceedings of the 1st ACM Workshop on Wireless Security, Aug. 2002, pp. 11–20.

[5] R. Horvath, D. Nedbal, and M. Stieninger, "A literature review on challenges and effects of software defined networking," Procedia Computer Science, vol. 64, pp. 552–561, 2015.

[6] R. Raghavendra, J. Lobo, and K.-W. Lee, "Dynamic graph query primitives for SDN-based cloud network management," in Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks, Aug. 2012, pp. 97–102.

[7] K. Hyojoon and N. Feamster, "Improving network management with software defined networking," IEEE Communications Magazine, vol. 51, no. 2, pp. 114–119, Feb. 2013.

[8] A. Lara and B. Ramamurthy, "OpenSec: Policy-based security using Software-Defined Networking," IEEE Transactions on Network and Service Management, vol. 13, no. 1, pp. 30–42, Mar. 2016.

[9] N. Zhisheng, W. Yiqun, G. Jie, and Y. Zexi, "Cell zooming for cost-efficient green cellular networks," IEEE Communications Magazine, vol. 48, no. 11, pp. 74–79, Nov. 2010.

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

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[12] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," IEEE Internet Computing, vol. 17, no. 2, pp. 20–27, Mar. 2013.