Independent submission
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

LM. Contreras
Telefonica
CJ. Bernardos
UC3M
D. Lopez
Telefonica
M. Boucadair
Orange
P. Iovanna
Ericsson
July 2, 2018

Cooperating Layered Architecture for SDN
draft-contreras-layered-sdn-02

Abstract

Software Defined Networking (SDN) proposes the separation of the control plane from the data plane in the network nodes and its logical centralization on a control entity. Most of the network intelligence is moved to this functional entity. Typically, such entity is seen as a compendium of interacting control functions in a vertical, tight integrated fashion. The relocation of the control functions from a number of distributed network nodes to a logical central entity conceptually places together a number of control capabilities with different purposes. As a consequence, the existing solutions do not provide a clear separation between transport control and services that relies upon transport capabilities.

This document describes a proposal named Cooperating Layered Architecture for SDN. The idea behind that is to differentiate the control functions associated to transport from those related to services, in such a way that they can be provided and maintained independently, and can follow their own evolution path.

Status of This Memo

Table of Contents

1. Introduction

Network softwarization advances are facilitating the introduction of programmability in services and infrastructures of telco operators. This is achieved generically through the introduction of Software Defined Networking (SDN) capabilities in the network, including controllers and orchestrators.

However, there are concerns of different nature that these SDN capabilities have to resolve. In one hand there is a need for actions focused on programming the network for handle the connectivity or forwarding of digital data between distant nodes. On the other hand, there is a need for actions devoted to program the functions or services that process (or manipulate) such digital data.

SDN proposes the separation of the control plane from the data plane in the network nodes by introducing abstraction among both planes, allowing to centralize the control logic on an entity which is commonly referred as SDN Controller. A programmatic interface is then defined between a forwarding entity (at the network node) and a central control entity. Through that interface, a control entity instructs the nodes involved in the forwarding plane and modifies their traffic forwarding behavior accordingly. Additional capabilities (e.g., performance monitoring, fault management, etc.) could be expected to be supported through such kind of programmatic interface [RFC7149].

Most of the intelligence is moved to such functional entity. Typically, such entity is seen as a compendium of interacting control functions in a vertical, tight integrated fashion.

The approach of considering an omnipotent control entity governing the overall aspects of a network, especially both the transport network and the services to be supported on top of it, presents a number of issues:

- From a provider perspective, where usually different departments are responsible of handling service and connectivity (i.e., transport capabilities for the service on top), the mentioned approach offers unclear responsibilities for complete service provision and delivery.

- Complex reuse of functions for the provision of services.

- Closed, monolithic control architectures.

- Difficult interoperability and interchangeability of functional components.

- Blurred business boundaries among providers, especially in situations where a provider provides just connectivity while another provider offers a more sophisticated service on top of that connectivity.

- Complex service/network diagnosis and troubleshooting, particularly to determine which segment is responsible for a failure.

The relocation of the control functions from a number of distributed network nodes to another entity conceptually places together a number of control capabilities with different purposes. As a consequence, the existing SDN solutions do not provide a clear separation between services and transport control. Here, the separation between service and transport follows the distinction provided by [Y.2011], and also defined in Section 2 of this document.

This document describes a proposal named Cooperating Layered Architecture for SDN (CLAS). The idea behind that is to differentiate the control functions associated to transport from those related to services, in such a way that they can be provided and maintained independently, and can follow their own evolution path.

Despite such differentiation it is required a close cooperation between service and transport layers (or strata in [Y.2011]) and associated components to provide an efficient usage of the resources.

2. Terminology

This document makes use of the following terms:

- Transport: denotes the transfer capabilities offered by a networking infrastructure. The transfer capabilities can rely upon pure IP techniques, or other means such as MPLS or optics.

- o Service: denotes a logical construct that makes use of transport capabilities. This document does not make any assumption on the functional perimeter of a service that can be built above a transport infrastructure. As such, a service can be an offering that is offered to customers or be invoked for the delivery of another (added-value) service.

- o Layer: refers to the set of elements comprised for enabling either transport or service capabilities as defined before. In [Y.2011], this is referred to as stratum, and both are used interchangeably.

- o Domain: is a set of elements which share a common property or characteristic. In this document this applies to administrative domain (i.e., elements pertaining to the same organization), technological domain (elements implementing the same kind of technology, as for example optical nodes), etc.

- o SDN intelligence: refers to the decision-making process that is hosted by a node or a set of nodes. The intelligence can be centralized or distributed. Both schemes are within the scope of this document. The SDN intelligence relies on inputs form various functional blocks such as: network topology discovery, service topology discovery, resource allocation, business guidelines, customer profiles, service profiles, etc. The exact decomposition of an SDN intelligence, apart from the layering discussed in this document, is out of scope.

Additionally, the following acronyms are used in this document.

CLAS: Cooperating Layered Architecture for SDN

FCAPS: Fault, Configuration, Accounting, Performance and Security

SDN: Software Defined Networking

SLA: Service Level Agreement

3. Architecture Overview

Current operator networks support multiple services (e.g., VoIP, IPTV, mobile VoIP, critical mission applications, etc.) on a variety of transport technologies. The provision and delivery of a service independently of the underlying transport capabilities require a separation of the service related functionalities and an abstraction of the transport network to hide the specificities of underlying transfer techniques while offering a common set of capabilities.

Such separation can provide configuration flexibility and adaptability from the point of view of either the services or the transport network. Multiple services can be provided on top of a common transport infrastructure, and similarly, different technologies can accommodate the connectivity requirements of a certain service. A close coordination among them is required for a consistent service delivery (inter-layer cooperation).

This document focuses particularly on:

o Means to expose transport capabilities to services.

o Means to capture service requirements of services.

o Means to notify service intelligence with underlying transport events, for example to adjust service decision-making process with underlying transport events.

o Means to instruct the underlying transport capabilities to accommodate new requirements, etc.

An example is to guarantee some Quality of Service (QoS) levels. Different QoS-based offerings could be present at both service and transport layers. Vertical mechanisms for linking both service and transport QoS mechanisms should be in place to provide the quality guarantees to the end user.

CLAS architecture assumes that the logically centralized control functions are separated in two functional layers. One of the functional layers comprises the service-related functions, whereas the other one contains the transport-related functions. The cooperation between the two layers is expected to be implemented through standard interfaces.

Figure 1 shows the CLAS architecture. It is based on functional separation in the NGN architecture defined by the ITU-T in [Y.2011], where two strata of functionality are defined, namely the Service Stratum, comprising the service-related functions, and the Connectivity Stratum, covering the transport ones. The functions on each of these layers are further grouped on control, management and user (or data) planes.

CLAS adopts the same structured model described in [Y.2011] but applying it to the objectives of programmability through SDN [RFC7149]. To this respect, CLAS proposes to address services and transport in a separated manner because of their differentiated concerns.

```
                                       Applications
                                            /\
                                            ||
                                            ||
      +-------------------------------------||-------------+
      | Service Stratum                     ||             |
      |                                     \/             |
      |                       ...........................  |
      |                       . SDN Controller          .  |
      |                       .                         .  |
      |  +--------------+     .        +--------------+ .  |
      |  | Resource Pl. |     .        |  Mngmt. Pl.  | .  |
      |  |              |<===>.  +--------------+     | .  |
      |  |              |     .  |  Control Pl. |     | .  |
      |  +--------------+     .  |              |-----+ .  |
      |                       .  |              |       .  |
      |                       .  +--------------+       .  |
      |                       ...........................  |
      |                                     /\             |
      |                                     ||             |
      +-------------------------------------||-------------+
                                            ||    Standard
                                         -- || --    API
                                            ||
      +-------------------------------------||-------------+
      | Transport Stratum                   ||             |
      |                                     \/             |
      |                       ...........................  |
      |                       . SDN Controller          .  |
      |                       .                         .  |
      |  +--------------+     .        +--------------+ .  |
      |  | Resource Pl. |     .        |  Mngmt. Pl.  | .  |
      |  |              |<===>.  +--------------+     | .  |
      |  |              |     .  |  Control Pl. |     | .  |
      |  +--------------+     .  |              |-----+ .  |
      |                       .  |              |       .  |
      |                       .  +--------------+       .  |
      |                       ...........................  |
      |                                                    |
      |                                                    |
      +----------------------------------------------------+
```

Figure 1: Cooperating Layered Architecture for SDN

In the CLAS architecture both the control and management functions are considered to be performed by one or a set of SDN controllers (due to, e.g., scalability, reliability), in such a way that separated SDN controllers are present in the Service and Transport

strata. Management functions are considered to be part of the SDN controller to allow the effective operation in a service provider ecosystem [RFC7149] despite some initial propositions did not consider such management as part of the SDN environment [ONFArch].

Furthermore, the generic user or data plane functions included in the NGN architecture are referred here as resource plane functions. The resource plane in each stratum is controlled by the corresponding SDN controller through a standard interface.

The SDN controllers cooperate for the provision and delivery of services. There is a hierarchy in which the Service SDN controller requests transport capabilities to the Transport SDN controller.

The Service SDN controller acts as a client of the Transport SDN controller.

Furthermore, the Transport SDN controller interacts with the Service SDN controller to inform it about events in the transport network that can motivate actions in the service layer.

Despite it is not shown in Figure 1, the resource planes of each stratum could be connected. This will depend on the kind of service provided. Furthermore, the Service stratum could offer an interface towards applications to expose network service capabilities to those applications or customers.

3.1. Functional Strata

As described before, the functional split separates transport-related functions from service-related functions. Both strata cooperate for a consistent service delivery.

Consistency is determined and characterized by the service layer.

3.1.1. Transport Stratum

The Transport Stratum comprises the functions focused on the transfer of data between the communication end points (e.g., between end-user devices, between two service gateways, etc.). The data forwarding nodes are controlled and managed by the Transport SDN component. The Control plane in the SDN controller is in charge of instructing the forwarding devices to build the end to end data path for each communication or to make sure forwarding service is appropriately setup. Forwarding may not be rely on the sole pre-configured entries; dynamic means can be enabled so that involved nodes can build dynamically routing and forwarding paths (this would require that the nodes retain some of the control and management capabilities

for enabling this). Finally, the Management plane performs management functions (i.e., FCAPS) on those devices, like fault or performance management, as part of the Transport Stratum capabilities.

3.1.2. Service Stratum

The Service stratum contains the functions related to the provision of services and the capabilities offered to external applications. The Resource plane consists of the resources involved in the service delivery, such as computing resources, registries, databases, etc. The Control plane is in charge of controlling and configuring those resources, as well as interacting with the Control plane of the Transport stratum in client mode for requesting transport capabilities for a given service. In the same way, the Management plane implements management actions on the service-related resources and interacts with the Management plane in the Transport Stratum for a cooperating management between layers.

3.1.3. Recursiveness

Recursive layering can happen in some usage scenarios in which the Transport Stratum is itself structured in Service and Transport Stratum. This could be the case of the provision of a transport service complemented with advanced capabilities additional to the pure data transport (e.g., maintenance of a given SLA [RFC7297]).

Recursiveness has been also discussed in [ONFArch] as a way of reaching scalability and modularity, when each higher level can provide greater abstraction capabilities. Additionally, recursiveness can allow some scenarios for multi-domain where single or multiple administrative domains are involved, as the ones described in Section 6.3.

3.2. Plane Separation

The CLAS architecture leverages on the SDN proposition of plane separation. As mentioned before, three different planes are considered for each stratum. The communication among these three planes (and with the corresponding plane in other strata) is based on open, standard interfaces.

3.2.1. Control Plane

The Control plane logically centralizes the control functions of each stratum and directly controls the corresponding resources. [RFC7426] introduces the role of the control plane in a SDN architecture. This plane is part of an SDN controller, and can interact with other

control planes in the same or different strata for accomplishing control functions.

3.2.2. Management Plane

The Management plane logically centralizes the management functions for each stratum, including the management of the Control and Resource planes. [RFC7426] describes the functions of the management plane in a SDN environment. This plane is also part of the SDN controller, and can interact with the corresponding management planes residing in SDN controllers of the same or different strata.

3.2.3. Resource Plane

The Resource plane comprises the resources for either the transport or the service functions. In some cases the service resources can be connected to the transport ones (e.g., being the terminating points of a transport function) whereas in other cases it can be decoupled from the transport resources (e.g., one database keeping some register for the end user). Both forwarding and operational planes proposed in [RFC7426] would be part of the Resource plane in this architecture.

4. Required Features

Since the CLAS architecture implies the interaction of different layers with different purposes and responsibilities, a number of features are required to be supported.

- o Abstraction: the mapping of physical resources into the corresponding abstracted resources.

- o Service parameter translation: translation of service parameters (e.g., in the form of SLAs) to transport parameters (or capabilities) according to different policies.

- o Monitoring: mechanisms (e.g. event notifications) available in order to dynamically update the (abstracted) resources' status taking in to account e.g. the traffic load.

- o Resource computation: functions able to decide which resources will be used for a given service request. As an example, functions like PCE could be used to compute/select/decide a certain path.

- o Orchestration: ability to combine diverse resources (e.g., IT and network resources) in an optimal way.

o Accounting: record of resource usage.

o Security: secure communication among components, preventing e.g. DoS attacks.

5. Communication Between SDN Controllers

The SDN controllers residing respectively in the Service and the Transport Stratum need to establish a tight coordination. Mechanisms for transfer relevant information for each stratum should be defined.

From the service perspective, the Service SDN controller needs to easily access transport resources through well-defined APIs to retrieve the capabilities offered by the Transport Stratum. There could be different ways of obtaining such transport-aware information, i.e., by discovering or publishing mechanisms. In the former case the Service SDN Controller could be able of handling complete information about the transport capabilities (including resources) offered by the Transport Stratum. In the latter case, the Transport Stratum exposes available capabilities e.g. through a catalog, reducing the amount of detail of the underlying network.

On the other hand, the Transport Stratum requires to properly capture Service requirements. These can include SLA requirements with specific metrics (such as delay), level of protection to be provided, max/min capacity, applicable resource constraints, etc.

The communication between controllers must be also secure, e.g. by preventing denial of service or any other kind of threats (similarly, the communications with the network nodes must be secure).

6. Deployment Scenarios

Different situations can be found depending on the characteristics of the networks involved in a given deployment.

6.1. Full SDN Environments

This case considers that the networks involved in the provision and delivery of a given service have SDN capabilities.

6.1.1. Multiple Service Strata Associated to a Single Transport Stratum

A single Transport Stratum can provide transfer functions to more than one Service strata. The Transport Stratum offers a standard interface(s) to each of the Service strata. The Service strata are the clients of the Transport Stratum. Some of the capabilities

offered by the Transport stratum can be isolation of the transport resources (slicing), independent routing, etc.

6.1.2. Single Service Stratum associated to multiple Transport Strata

A single Service stratum can make use of different Transport Strata for the provision of a certain service. The Service stratum interfaces each of the Transport Strata with standard protocols, and orchestrates the provided transfer capabilities for building the end to end transport needs.

6.2. Hybrid Environments

This case considers scenarios where one of the strata is legacy totally or in part.

6.2.1. SDN Service Stratum associated to a Legacy Transport Stratum

An SDN service stratum can interact with a legacy Transport Stratum through some interworking function able to adapt SDN-based control and management service-related commands to legacy transport-related protocols, as expected by the legacy Transport Stratum. The SDN controller in the Service stratum is not aware of the legacy nature of the underlying Transport Stratum.

6.2.2. Legacy Service Stratum Associated to an SDN Transport Stratum

A legacy Service stratum can work with an SDN-enabled Transport Stratum through the mediation of and interworking function capable to interpret commands from the legacy service functions and translate them into SDN protocols for operating with the SDN-enabled Transport Stratum.

6.3. Multi-domain Scenarios in Transport Stratum

The Transport Stratum can be composed by transport resources being part of different administrative, topological or technological domains. The Service Stratum can yet interact with a single entity in the Transport Stratum in case some abstraction capabilities are provided in the transport part to emulate a single stratum.

Those abstraction capabilities constitute a service itself offered by the Transport Stratum to the services making use of it. This service is focused on the provision of transport capabilities, then different of the final communication service using such capabilities.

In this particular case this recursion allows multi-domain scenarios at transport level.

Multi-domain situations can happen in both single-operator and multi-operator scenarios.

In single operator scenarios a multi-domain or end-to-end abstraction component can provide an homogeneous abstract view of the underlying heterogeneous transport capabilities for all the domains.

Multi-operator scenarios, at the Transport Stratum, should support the establishment of end-to-end paths in a programmatic manner across the involved networks. This could be accomplished e.g. by the exchange of traffic-engineered information of each of the administrative domains [RFC7926].

7. Use cases

This section presents a number of use cases as examples of applicability of the CLAS proposal

7.1. Network Function Virtualization

NFV environments offer two possible levels of SDN control [ETSI_NFV_EVE005]. One level is the need for controlling the NFVI to provide connectivity end-to- end among VNFs (Virtual Network Functions) or among VNFs and PNFs (Physical Network Functions). A second level is the control and configuration of the VNFs themselves (in other words, the configuration of the network service implemented by those VNFs), taking profit of the programmability brought by SDN. Both control concerns are separated in nature. However, interaction between both could be expected in order to optimize, scale or influence each other.

7.2. Abstraction and Control of Transport Networks

Abstraction and Control of Transport Networks (ACTN) [I-D.ietf-teas-actn-framework] presents a framework to allow the creation of virtual networks to be offered to customers. The concept of provider in ACTN is limited to the offering of virtual network services. These services are essentially transport services, and would correspond to the Transport Stratum in CLAS. On the other hand, the Service Stratum in CLAS can be assimilated as a customer in the context of ACTN.

ACTN propose a hierarchy of controllers for facilitating the creation and operation of the virtual networks. An interface is proposed for the relation of the customers requesting these virtual networks services with the controller in charge of orchestrating and serving such request. Such interface is equivalent to the one defined in Figure 1 of this document between Service and Transport Strata.

8. Challenges for Implementing Actions Between Service and Transport Strata

The distinction of service and transport concerns raises a number of challenges in the communication between both strata. The following is a work-in-progress list reflecting some of the identified challenges:

- Standard mechanisms for interaction between layers: Nowadays there are a number of proposals that could accommodate requests from the service stratum to the transport stratum. Some of them could be solutions like the Connectivity Provisioning Protocol [I-D.boucadair-connectivity-provisioning-protocol] or the Intermediate-Controller Plane Interface (I-CPI) [ONFArch]. Other potential candidates could be the Transport API [TAPI] or the Transport NBI [I-D.tnbidt-ccamp-transport-nbi-use-cases]. Each of these options has a different status of maturity and scope.

- Multi-provider awareness: In multi-domain scenarios involving more than one provider at transport level, the service stratum could have or not awareness of such multiplicity of domains. If the service stratum is unaware of the multi-domain situation, then the Transport Stratum acting as entry point of the service stratum request should be responsible of managing the multi-domain issue. On the contrary, if the service stratum is aware of the multi-domain situation, it should be in charge of orchestrating the requests to the different underlying Transport Strata for composing the final end-to-end path among service end-points (i.e., functions).

- SLA mapping: Both strata will handle SLAs but the nature of those SLAs could differ. Then it is required for the entities in each stratum to map service SLAs to connectivity SLAs in order to ensure proper service delivery.

- Association between strata: The association between strata could be configured beforehand, or could be dynamic following mechanisms of discovery, that could be required to be supported by both strata with this purpose.

- Security: As reflected before, the communication between strata must be secure preventing attacks and threats. Additionally, privacy should be enforced, especially when addressing multi-provider scenarios at transport level.

- Accounting: The control and accountancy of resources used and consumed by services should be supported in the communication among strata.

9. IANA Considerations

This document does not request any action from IANA.

10. Security Considerations

The CLAS architecture relies upon the functional entities that are introduced in [RFC7149] and [RFC7426]. As such security considerations discussed in Section 5 of [RFC7149], in particular, must be taken into account.

The communication between the service and transport SDN controllers must rely on secure means which achieve the following:

- o Mutual authentication must be enabled before taking any action.

- o Message integrity protection.

Each of the controllers must be provided with instructions about the set of information (and granularity) that can be disclosed to a peer controller. Means to prevent leaking privacy data (e.g., from the service stratum to the transport stratum) must be enabled. The exact set of information to be shared is deployment-specific.

A corrupted controller may induce some disruption on another controller. Guards against such attacks should be enabled.

Security in the communication between the strata here described should apply on the APIs (and/or protocols) to be defined among them. In consequence, security concerns will correspond to the specific solution.

11. Acknowledgements

This document was previously discussed and adopted in the IRTF SDN RG as [I-D.irtf-sdnrg-layered-sdn]. After the closure of the IRTF SDN RG this document is being progressed as Individual Submission to record (some of) that group's disucussions.

The authors would like to thank (in alphabetical order) Bartosz Belter, Gino Carrozzo, Ramon Casellas, Gert Grammel, Ali Haider, Evangelos Haleplidis, Zheng Haomian, Giorgios Karagianis, Gabriel Lopez, Maria Rita Palatella, Christian Esteve Rothenberg and Jacek Wytrebowicz for their comments and suggestions.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[Y.2011] "General principles and general reference model for Next Generation Networks", ITU-T Recommendation Y.2011 , October 2004.

12.2. Informative References

[ETSI_NFV_EVE005]
"Report on SDN Usage in NFV Architectural Framework", December 2015.

[I-D.boucadair-connectivity-provisioning-protocol]
Boucadair, M., Jacquenet, C., Zhang, D., and P. Georgatsos, "Connectivity Provisioning Negotiation Protocol (CPNP)", draft-boucadair-connectivity-provisioning-protocol-15 (work in progress), December 2017.

[I-D.ietf-teas-actn-framework]
Ceccarelli, D. and Y. Lee, "Framework for Abstraction and Control of Traffic Engineered Networks", draft-ietf-teas-actn-framework-15 (work in progress), May 2018.

[I-D.irtf-sdnrg-layered-sdn]
Contreras, L., Bernardos, C., Lopez, D., Boucadair, M., and P. Iovanna, "Cooperating Layered Architecture for SDN", draft-irtf-sdnrg-layered-sdn-01 (work in progress), October 2016.

[I-D.tnbidt-ccamp-transport-nbi-use-cases]
Busi, I. and D. King, "Transport Northbound Interface Applicability Statement and Use Cases", draft-tnbidt-ccamp-transport-nbi-use-cases-03 (work in progress), September 2017.

[ONFArch] Open Networking Foundation, "SDN Architecture, Issue 1", June 2014, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <https://www.rfc-editor.org/info/rfc7149>.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, DOI 10.17487/RFC7297, July 2014, <https://www.rfc-editor.org/info/rfc7297>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G., Ceccarelli, D., and X. Zhang, "Problem Statement and Architecture for Information Exchange between Interconnected Traffic-Engineered Networks", BCP 206, RFC 7926, DOI 10.17487/RFC7926, July 2016, <https://www.rfc-editor.org/info/rfc7926>.

[TAPI] "Functional Requirements for Transport API", June 2016.

Appendix A. Relationship with RFC7426

[RFC7426] introduces an SDN taxonomy by defining a number of planes, abstraction layers, and interfaces or APIs among them, as a means of clarifying how the different parts constituent of SDN (network devices, control and management) relate among them. A number of planes are defined, namely:

o Forwarding Plane: focused on delivering packets in the data path based on the instructions received from the control plane.

o Operational Plane: centered on managing the operational state of the network device.

o Control Plane: devoted to instruct the device on how packets should be forwarded.

o Management Plane: in charge of monitoring and maintaining network devices.

o Application Plane: enabling the usage for different purposes (as determined by each application) of all the devices controlled in this manner.

Apart from that, [RFC7426] proposes a number of abstraction layers that permit the integration of the different planes through common interfaces. CLAS focuses on Control, Management and Resource planes as the basic pieces of its architecture. Essentially, the control plane modifies the behavior and actions of the controlled resources. The management plane monitors and retrieves the status of those resources. And finally, the resource plane groups all the resources related to the concerns of each strata.

From this point of view, CLAS planes can be seen as a superset of [RFC7426], even though in some cases not all the planes as considered in [RFC7426] could not be totally present in CLAS representation (e.g., forwarding plane in Service Stratum).

Being said that, internal structure of CLAS strata could follow the taxonomy defined in [RFC7426]. Which is differential is the specialization of the SDN environments, through the distinction between service and transport.

Authors' Addresses

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: http://lmcontreras.com

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: http://www.it.uc3m.es/cjbc/

Diego R. Lopez
Telefonica
Ronda de la Comunicacion, s/n
Sur-3 building, 3rd floor
Madrid 28050
Spain

Email: diego.r.lopez@telefonica.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Paola Iovanna
Ericsson
Pisa
Italy

Email: paola.iovanna@ericsson.com