Independent Submission
Request for Comments: 8597
Category: Informational
ISSN: 2070-1721

LM. Contreras
Telefonica
CJ. Bernardos
UC3M
D. Lopez
Telefonica
M. Boucadair
Orange
P. Iovanna
Ericsson

May 2019

Cooperating Layered Architecture for Software-Defined Networking (CLAS)

Abstract

Software-Defined Networking (SDN) advocates for the separation of the control plane from the data plane in the network nodes and its logical centralization on one or a set of control entities. Most of the network and/or service intelligence is moved to these control entities. Typically, such an entity is seen as a compendium of interacting control functions in a vertical, tightly 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 rely upon transport capabilities.

This document describes an approach called Cooperating Layered Architecture for Software-Defined Networking (CLAS), wherein the control functions associated with transport are differentiated 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

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8597.

Copyright Notice

Table of Contents

1. Introduction

Network softwarization advances are facilitating the introduction of programmability in the services and infrastructures of telecommunications operators. This is generally achieved through the introduction of Software-Defined Networking (SDN) [RFC7149] [RFC7426] capabilities in the network, including controllers and orchestrators.

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

SDN advocates for the separation of the control plane from the data plane in the network nodes by introducing abstraction among both planes, allowing the control logic on a functional entity, which is commonly referred as SDN Controller, to be centralized; one or multiple controllers may be deployed. A programmatic interface is then defined between a forwarding entity (at the network node) and a control entity. Through that interface, a control entity instructs the nodes involved in the forwarding plane and modifies their traffic-forwarding behavior accordingly. Support for additional capabilities (e.g., performance monitoring, fault management, etc.) could be expected through this kind of programmatic interface [RFC7149].

Most of the intelligence is moved to this kind of functional entity. Typically, such an entity is seen as a compendium of interacting control functions in a vertical, tightly 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 different departments usually are responsible for 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 one provider provides only connectivity while another provider offers a more sophisticated service on top of that connectivity.

- Complex service/network diagnosis and troubleshooting, particularly to determine which layer 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 as defined in Section 2 of this document.

This document describes an approach called Cooperating Layered Architecture for SDN (CLAS), wherein the control functions associated with transport are differentiated 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, close cooperation between the service and transport layers (or strata in [Y.2011]) and the associated components are necessary to provide 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.

- Service: denotes a logical construct that makes use of transport capabilities.

  This document does not make any assumptions about the functional perimeter of a service that can be built above a transport infrastructure. As such, a service can be offered to customers or invoked for the delivery of another (added-value) service.

- Layer: refers to the set of elements that enable either transport or service capabilities, as defined previously. In [Y.2011], this is referred to as a "stratum", and the two terms are used interchangeably.

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

- SDN Intelligence: refers to the decision-making process that is hosted by a node or a set of nodes. These nodes are called SDN controllers.

  The intelligence can be centralized or distributed. Both schemes are within the scope of this document.

  An SDN Intelligence relies on inputs from 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 here, is out of the scope of this document.

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., Voice over IP (VoIP), IPTV, mobile VoIP, critical mission applications, etc.) on a variety of transport technologies. The provision and delivery of a service independent of the underlying transport capabilities require a separation of the service-related functionalities and an abstraction of the transport network to hide the specifics of the 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; similarly, different technologies can accommodate the connectivity requirements of a certain service. Close coordination among these elements is required for consistent service delivery (inter-layer cooperation).

This document focuses particularly on the means to:

o expose transport capabilities to services.

o capture transport requirements of services.

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

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

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

CLAS architecture assumes that the logically centralized control functions are separated into 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 Next Generation Network (NGN) architecture defined by the ITU-T in [Y.2011], where two strata of functionality are defined. These strata are the Service Stratum, comprising the service-related functions, and the Transport Stratum, covering the transport-related functions. The functions of each of these layers are further grouped into the control, management, and user (or data) planes.

CLAS adopts the same structured model described in [Y.2011] but applies it to the objectives of programmability through SDN [RFC7149]. In this respect, CLAS advocates for addressing services and transport in a separated manner because of their differentiated concerns.

```
                                       Applications
                                            /\
                                            ||
                                            ||
      +-------------------------------------||-------------+
      | Service Stratum                     ||             |
      |                                     \/             |
      |                       ...........................  |
      |                       . SDN Intelligence        .  |
      |                       .                         .  |
      |  +--------------+     .        +--------------+ .  |
      |  | Resource Pl. |     .        |   Mgmt. Pl.  | .  |
      |  |              |<===>.  +--------------+     | .  |
      |  |              |     .  |  Control Pl. |     | .  |
      |  +--------------+     .  |              |-----+ .  |
      |                       .  |              |       .  |
      |                       .  +--------------+       .  |
      |                       ...........................  |
      |                                     /\             |
      |                                     ||             |
      +-------------------------------------||-------------+
                                            ||    Standard
                                         -- || --    API
                                            ||
      +-------------------------------------||-------------+
      | Transport Stratum                   ||             |
      |                                     \/             |
      |                       ...........................  |
      |                       . SDN Intelligence        .  |
      |                       .                         .  |
      |  +--------------+     .        +--------------+ .  |
      |  | Resource Pl. |     .        |   Mgmt. 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, for example, scalability, reliability), providing the SDN Intelligence 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 Intelligence to allow for effective operation in a service provider ecosystem [RFC7149], although 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 to here as resource-plane functions. The resource plane in each stratum is controlled by the corresponding SDN Intelligence through a standard interface.

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

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

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

Despite not being 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 to applications to expose network service capabilities to those applications or customers.

3.1. Functional Strata

As aforementioned, there is a functional split that separates transport-related functions from service-related functions. Both strata cooperate for 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 endpoints (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 Intelligence is in charge of instructing the forwarding devices to build the end-to-end data path for each communication or to make sure the forwarding service is appropriately set up. Forwarding may not be rely solely on the preconfigured entries; means can be enabled so that involved nodes can dynamically build 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 to request 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 to ensure management cooperation between layers.

3.1.3. Recursiveness

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

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

3.2. Plane Separation

The CLAS architecture leverages plane separation. As mentioned in Sections 3.1.1 and 3.1.2, three different planes are considered for each stratum. The communication among these three planes (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 an SDN architecture. This plane is part of an SDN Intelligence and can interact with other control planes in the same or different strata to perform 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 an SDN environment. This plane is also part of the SDN Intelligence 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); in other cases, it can be decoupled from the transport resources (e.g., one database keeping a register for the end user). Both the 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:

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

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

- Monitoring: mechanisms (e.g., event notifications) available in order to dynamically update the (abstracted) resources' status while taking into account, for example, 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: the 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, for example, DoS attacks.

5. Communication between SDN Controllers

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

From the service perspective, the Service SDN Intelligence 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 Intelligence could be able to handle complete information about the transport capabilities (including resources) offered by the Transport Stratum. In the latter case, the Transport Stratum reveals the available capabilities, for example, through a catalog, reducing the amount of detail of the underlying network.

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

The communication between controllers must also be secure, e.g., by preventing denial of service or any other kind of threat (similarly, 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 with a Single Transport Stratum

A single Transport Stratum can provide transfer functions to more than one Service Stratum. 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 with Multiple Transport Strata

A single Service Stratum can make use of different Transport Strata for the provision of a certain service. The Service Stratum invokes standard interfaces to each of the Transport Strata, 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 totally or partly legacy.

6.2.1. SDN Service Stratum Associated with a Legacy Transport Stratum

An SDN service Stratum can interact with a legacy Transport Stratum through an interworking function that is 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 Intelligence in the Service Stratum is not aware of the legacy nature of the underlying Transport Stratum.

6.2.2. Legacy Service Stratum Associated with an SDN Transport Stratum

A legacy Service Stratum can work with an SDN-enabled Transport Stratum through the mediation of an interworking function capable of interpreting commands from the legacy service functions and translating them into SDN protocols for operation with the SDN-enabled Transport Stratum.

6.3. Multi-domain Scenarios in the Transport Stratum

The Transport Stratum can be composed of transport resources that are part of different administrative, topological, or technological domains. The Service Stratum can 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 this stratum. This service is focused on the provision of transport capabilities, which is different from the final communication service using such capabilities.

In this particular case, this recursion allows multi-domain scenarios at the 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 a 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. For example, this could be accomplished by each of the administrative domains exchanging their traffic-engineered information [RFC7926].

7. Use Cases

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

7.1. Network Function Virtualization (NFV)

NFV environments offer two possible levels of SDN control [GSNFV-EVE005]. One level is the need to control the NFV Infrastructure (NFVI) to provide end-to-end connectivity 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), which benefits from the programmability brought by SDN. The two control concerns are separate in nature. However, interaction between the two can be expected in order to optimize, scale, or influence one another.

7.2. Abstraction and Control of TE Networks

Abstraction and Control of TE Networks (ACTN) [RFC8453] presents a framework that allows 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 defines a hierarchy of controllers to facilitate the creation and operation of the virtual networks. An interface is defined for the relationship between the customers requesting these virtual network services and the controller in charge of orchestrating and serving such a request. Such an interface is equivalent to the one defined in Figure 1 (Section 3) between the 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 the two strata. The following list reflects 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 the proposals could be solutions like the Connectivity Provisioning Negotiation Protocol [CPNP] or the Intermediate-Controller Plane Interface (I-CPI) [ONFArch].

  Other potential candidates could be the Transport API [TAPI] or the Transport Northbound Interface [TRANS-NORTH]. Each of these options has a different scope.

- Multi-provider awareness: In multi-domain scenarios involving more than one provider at the transport level, the Service Stratum may or may not be aware of such multiplicity of domains.

  If the Service Stratum is unaware of the multi-domain situation, then the Transport Stratum acting as the entry point of the Service Stratum request should be responsible for 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 to compose the final end-to-end path among service endpoints (i.e., service functions).

- SLA mapping: Both strata will handle SLAs, but the nature of those SLAs could differ. Therefore, 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 both strata could require the use of a discovery mechanism that dynamically establishes the association between the strata.

- Security: As reflected before, the communication between strata must be secure to prevent attacks and threats. Additionally, privacy should be enforced, especially when addressing multi-provider scenarios at the 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 has no IANA actions.

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 that achieve the following:

- Mutual authentication must be enabled before taking any action.

- Message integrity protection.

Each of the controllers must be provided with instructions regarding the set of information (and granularity) that can be disclosed to a peer controller. Means to prevent the leaking of 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. Protection against such attacks should be enabled.

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

11. References

11.1. Normative References

[Y.2011] International Telecommunication Union, "General principles and general reference model for Next Generation Networks", ITU-T Recommendation Y.2011, October 2004, <https://www.itu.int/rec/T-REC-Y.2011-200410-I/en>.

11.2. Informative References

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

[GSNFV-EVE005] ETSI, "Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework", ETSI GS NFV-EVE 005, V1.1.1, December 2015, <https://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_nfv-eve005v010101p.pdf>.

[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>.

[RFC8453] Ceccarelli, D., Ed. and Y. Lee, Ed., "Framework for Abstraction and Control of TE Networks (ACTN)", RFC 8453, DOI 10.17487/RFC8453, August 2018, <https://www.rfc-editor.org/info/rfc8453>.

[SDN-ARCH] Contreras, LM., Bernardos, CJ., Lopez, D., Boucadair, M., and P. Iovanna, "Cooperating Layered Architecture for SDN", Work in Progress, draft-irtf-sdnrg-layered-sdn-01, October 2016.

[TAPI] Open Networking Foundation, "Functional Requirements for Transport API", June 2016, <https://www.opennetworking.org/wp-content/uploads/2014/10/TR-527_TAPI_Functional_Requirements.pdf>.

[TRANS-NORTH] Busi, I., King, D., Zheng, H., and Y. Xu, "Transport Northbound Interface Applicability Statement", Work in Progress, draft-ietf-ccamp-transport-nbi-app-statement-05, March 2019.

Appendix A. Relationship with RFC 7426

[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. A number of planes are defined, including:

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

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

- Control Plane: dedicated to instructing the device on how packets should be forwarded.

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

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

Apart from these, [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 stratum.

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

That being said, the internal structure of CLAS strata could follow the taxonomy defined in [RFC7426]. What is different is the specialization of the SDN environments through the distinction between service and transport.

Acknowledgements

This document was previously discussed and adopted in the IRTF SDN RG as [SDN-ARCH]. After the closure of the IRTF SDN RG, this document was progressed as an Independent Submission to record (some of) that group's discussions.

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.

Thanks to Adrian Farrel for the review.

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