Internet Engineering Task Force (IETF)
Request for Comments: 7149
Category: Informational
ISSN: 2070-1721
March 2014

M. Boucadair
C. Jacquenet
France Telecom

Software-Defined Networking: A Perspective from within a Service Provider Environment

Abstract

Software-Defined Networking (SDN) has been one of the major buzz words of the networking industry for the past couple of years. And yet, no clear definition of what SDN actually covers has been broadly admitted so far. This document aims to clarify the SDN landscape by providing a perspective on requirements, issues, and other considerations about SDN, as seen from within a service provider environment.

It is not meant to endlessly discuss what SDN truly means but rather to suggest a functional taxonomy of the techniques that can be used under an SDN umbrella and to elaborate on the various pending issues the combined activation of such techniques inevitably raises. As such, a definition of SDN is only mentioned for the sake of clarification.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Internet has become the federative network that supports a wide range of service offerings. The delivery of network services such as IP VPNs assumes the combined activation of various capabilities that include (but are not necessarily limited to) forwarding and routing (e.g., customer-specific addressing scheme management, dynamic path computation to reach a set of destination prefixes, dynamic establishment of tunnels, etc.); Quality of Service (e.g., traffic classification, marking, conditioning, and scheduling); security (e.g., filters to protect customer premises from network-originated attacks, to avoid malformed route announcements, etc.); and management (e.g., fault detection and processing).

As these services not only grow in variety but also in complexity, their design, delivery, and operation have become a complex alchemy that often requires various levels of expertise. This situation is further aggravated by the wide variety of (network) protocols and tools, as well as recent convergence trends driven by Any Time, Any Where, Any Device (ATAWAD); ATAWADs are meant to make sure that an end user can access the whole range of services he/she has subscribed to whatever the access and device technologies, wherever the end user is connected to the network, and whether or not this end user is in motion.

Yet, most of these services have been deployed for the past decade, primarily based upon often static service production procedures that are more and more exposed to the risk of erroneous configuration commands. In addition, most of these services do not assume any specific negotiation between the customer and the service provider or between service providers, besides the typical financial terms.

At best, five-year master plans are referred to as the network planning policy that will be enforced by the service provider given the foreseen business development perspectives, manually computed traffic forecasts, and market coverage (fixed/mobile and residential/corporate). This so-called network planning policy may very well affect the way resources are allocated in a network, but it clearly fails to be adequately responsive to highly dynamic customer requirements in an "always-on" fashion. The need for improved service delivery procedures (including the time it takes to deliver the service once the possible negotiation phase is completed) is even more critical for corporate customers.

In addition, various tools are used for different, sometimes service-centric, management purposes, but their usage is not necessarily coordinated for event aggregation, correlation, and processing. This lack of coordination may come at the cost of extra complexity and possible customer Quality-of-Experience degradation.

Multi-service, multi-protocol, multi-technology-convergent, and dynamically adaptive networking environments of the near future have therefore become one of the major challenges faced by service providers.

This document aims to clarify the SDN landscape by providing a perspective on the functional taxonomy of the techniques that can be used in SDN, as seen from within a service provider environment.

2. Introducing Software-Defined Networking

2.1. A Tautology?

The separation of the forwarding and control planes (beyond implementation considerations) has almost become a gimmick to promote flexibility as a key feature of the SDN approach. Technically, most of the current router implementations have been assuming this separation for decades. Routing processes (such as IGP and BGP route computation) have often been software based, while forwarding capabilities are usually implemented in hardware.

As such, at the time of writing, what is considered to be state of the art tends to confirm the said separation, which rather falls under a tautology.

But, a somewhat centralized, "controller-embedded", control plane for the sake of optimized route computation before the Forwarding Information Base (FIB) population is certainly another story.

2.2. On Flexibility

Promoters of SDN have argued that it provides additional flexibility in how the network is operated. This is undoubtedly one of the key objectives that must be achieved by service providers. This is because the ability to dynamically adapt to a wide range of customer requests for flexible network service delivery is an important competitive advantage. But, flexibility is much, much more than separating the control and forwarding planes to facilitate forwarding decision-making processes.

For example, the ability to accommodate short duration extra bandwidth requirements so that end users can stream a video file to their 4G terminal device is an example of the flexibility that several mobile operators are currently investigating.

From this perspective, the ability to predict the network behavior as a function of the network services to be delivered is of paramount importance for service providers, so that they can assess the impact of introducing new services or activating additional network features or enforcing a given set of (new) policies from both financial and technical standpoints. This argues in favor of investigating advanced network emulation engines, which can be fed with information that can be derived from [LS-DISTRIB], for example.

Given the rather broad scope that the term "flexibility" suggests:

- Current SDN-labeled solutions are claimed to be flexible, although the notion is hardly defined. The exact characterization of what flexibility actually means is yet to be provided. Further work needs, therefore, to be conducted so that flexibility can be precisely defined in light of various criteria such as network evolution capabilities as a function of the complexity introduced by the integration of SDN techniques and seamless capabilities (i.e., the ability to progressively introduce SDN-enabled devices without disrupting network and service operation, etc.).

- The exposure of programmable interfaces is not a goal per se; rather, it is a means to facilitate configuration procedures for improved flexibility.

2.3. A Tentative Definition

We define Software-Defined Networking as the set of techniques used to facilitate the design, delivery, and operation of network services in a deterministic, dynamic, and scalable manner. The said determinism refers to the ability to completely master the various components of the service delivery chain, so that the service that has been delivered complies with what has been negotiated and contractually defined with the customer.

As such, determinism implies that the ability to control how network services are structured, designed, and delivered and where traffic should be forwarded in the network is for optimized resource usage. Although not explicitly restated in the following sections of the document, determinism lies beneath any action that may be taken by a service provider once service parameter negotiation is completed, from configuration tasks to service delivery, fulfillment, and assurance (see Section 2.4 below).

Such a definition assumes the introduction of a high level of automation in the overall service delivery and operation procedures.

Because networking is software driven by nature, the above definition does not emphasize the claimed "software-defined" properties of SDN-labeled solutions.

2.4. Functional Metadomains

SDN techniques can be classified into the following functional metadomains:

- Techniques for the dynamic discovery of network topology, devices, and capabilities, along with relevant information and data models that are meant to precisely document such topology, devices, and their capabilities.

- Techniques for exposing network services and their characteristics and for dynamically negotiating the set of service parameters that will be used to measure the level of quality associated with the delivery of a given service or a combination thereof. An example of this can be seen in [CPP].

- Techniques used by service-requirement-derived dynamic resource allocation and policy enforcement schemes, so that networks can be programmed accordingly. Decisions made to dynamically allocate resources and enforce policies are typically the result of the correlation of various inputs, such as the status of available resources in the network at any given time, the number of customer service subscription requests that need to be processed over a given period of time, the traffic forecasts, the possible need to trigger additional resource provisioning cycles according to a typical multi-year master plan, etc.

- Dynamic feedback mechanisms that are meant to assess how efficiently a given policy (or a set thereof) is enforced from a service fulfillment and assurance perspective.

3. Reality Check

The networking ecosystem has become awfully complex and highly demanding in terms of robustness, performance, scalability, flexibility, agility, etc. This means, in particular, that service providers and network operators must deal with such complexity and operate networking infrastructures that can evolve easily, remain scalable, guarantee robustness and availability, and are resilient to denial-of-service attacks.

The introduction of new SDN-based networking features should obviously take into account this context, especially from a cost impact assessment perspective.

3.1. Remember the Past

SDN techniques are not the next big thing per se but rather a kind of rebranding of proposals that have been investigated for several years, like active or programmable networks [AN] [PN]. As a matter of fact, some of the claimed "new" SDN features have been already implemented (e.g., Network Management System (NMS) and Path Computation Element (PCE) [RFC4655]) and supported by vendors for quite some time.

Some of these features have also been standardized (e.g., DNS-based routing [RFC1383]) that can be seen as an illustration of separated control and forwarding planes or Forwarding and Control Element Separation (ForCES) [RFC5810] [RFC5812].

Also, the policy-based management framework [RFC2753] introduced in the early 2000’s was designed to orchestrate available resources by means of a typical Policy Decision Point (PDP), which masters advanced offline traffic engineering capabilities. As such, this framework has the ability to interact with in-band software modules embedded in controlled devices (or not).

PDP is where policy decisions are made. PDPs use a directory service for policy repository purposes. The policy repository stores the policy information that can be retrieved and updated by the PDP. The PDP delivers policy rules to the Policy Enforcement Point (PEP) in the form of policy-provisioning information that includes configuration information.

PEP is where policy decisions are applied. PEPs are embedded in (network) devices, which are dynamically configured based upon the policy-formatted information that has been processed by the PEP. PEPs request configuration from the PDP, store the configuration information in the Policy Information Base (PIB), and delegate any policy decision to the PDP.

SDN techniques as a whole are an instantiation of the policy-based management framework. Within this context, SDN techniques can be used to activate capabilities on demand, to dynamically invoke network and storage resources, and to operate dynamically adaptive networks according to events (e.g., alteration of the network topology), triggers (e.g., dynamic notification of a link failure), etc.

3.2. Be Pragmatic

SDN approaches should be holistic, i.e., global and network wide. It is not a matter of configuring devices one by one to enforce a specific forwarding policy. Instead, SDN techniques are about configuring and operating a whole range of devices at the scale of the network for automated service delivery [AUTOMATION], from service negotiation (e.g., [CPNP]) and creation (e.g., [SLA-EXCHANGE]) to assurance and fulfillment.

Because the complexity of activating SDN capabilities is largely hidden from the end user and is software handled, a clear understanding of the overall ecosystem is needed to figure out how to manage this complexity and to what extent this hidden complexity does not have side effects on network operation.

As an example, SDN designs that assume a central decision-making entity must avoid single points of failure. They must not affect packet forwarding performances either (e.g., transit delays must not be impacted).

SDN techniques are not necessary to develop new network services per se. The basic service remains as (IP) connectivity that solicits resources located in the network. SDN techniques can thus be seen as another means to interact with network service modules and invoke both connectivity and storage resources accordingly in order to meet service-specific requirements.

By definition, SDN technique activation and operation remain limited to what is supported by embedded software and hardware. One cannot expect SDN techniques to support unlimited customizable features.

3.3. Measure Experience against Expectations

Because several software modules may be controlled by external entities (typically, a PDP), there is a need for a means to make sure that what has been delivered complies with what has been negotiated. Such means belong to the set of SDN techniques.

These typical policy-based techniques should interact with both Service Structuring engines (that are meant to expose the service characteristics and possibly negotiate those characteristics) and the network to continuously assess whether the experienced network behavior is compliant with the objectives set by the Service Structuring engine and those that may have been dynamically negotiated with the customer (e.g., as captured in a CPP [CPP] [CPNP]). This requirement applies to several regions of a network, including:

1. At the interface between two adjacent IP network providers.
2. At the access interface between a service provider and an IP network provider.
3. At the interface between a customer and the IP network provider.

Ideally, a fully automated service delivery procedure, from negotiation, ordering, and order processing to delivery, assurance, and fulfillment, should be supported at the cost of implications that are discussed in Section 4.1. This approach also assumes widely adopted standard data and information models in addition to interfaces.

3.4. Design Carefully

Exposing open and programmable interfaces has a cost from both scalability and performance standpoints.

Maintaining hard-coded performance optimization techniques is encouraged. So is the use of interfaces that allow the direct control of some engines (e.g., routing and forwarding) without requiring any in-between adaptation layers (generic objects to vendor-specific command line interfaces (CLIs), for instance). Nevertheless, the use of vendor-specific access means to some engines that it could be beneficial from a performance standpoint, at the cost of increasing the complexity of configuration tasks.

SDN techniques will have to accommodate vendor-specific components anyway. Indeed, these vendor-specific features will not cease to exist mainly because of the harsh competition.

The introduction of new functions or devices that may jeopardize network flexibility should be avoided or at least carefully considered in light of possible performance and scalability impacts. SDN-enabled devices will have to coexist with legacy systems.

One single SDN network-wide deployment is, therefore, very unlikely. Instead, multiple instantiations of SDN techniques will be progressively deployed and adapted to various network and service segments.

3.5. On OpenFlow

Empowering networking with in-band controllable modules may rely upon the OpenFlow protocol but also use other protocols to exchange information between a control plane and a data plane.

Indeed, there are many other candidate protocols that can be used for the same or even a broader purpose (e.g., resource reservation purposes). The forwarding of the configuration information can, for example, rely upon protocols like the Path Computation Element (PCE) Communication Protocol (PCEP) [RFC5440], the Network Configuration Protocol (NETCONF) [RFC6241], COPS Usage for Policy Provisioning (COPS-PR) [RFC3084], Routing Policy Specification Language (RPSL) [RFC2622], etc.

There is, therefore, no 1:1 relationship between OpenFlow and SDN. Rather, OpenFlow is one of the candidate protocols to convey specific configuration information towards devices. As such, OpenFlow is one possible component of the global SDN toolkit.

3.6. Non-goals

There are inevitable trade-offs to be found between operating the current networking ecosystem and introducing some SDN techniques, possibly at the cost of introducing new technologies. Operators do not have to choose between the two as both environments will have to coexist.

In particular, the following considerations cannot justify the deployment of SDN techniques:

- Fully flexible software implementations because the claimed flexibility remains limited by the software and hardware limitations, anyway.

- Fully modular implementations are difficult to achieve (because of the implicit complexity) and may introduce extra effort for testing, validation, and troubleshooting.

- Fully centralized control systems that are likely to raise some scalability issues. Distributed protocols and their ability to react to some events (e.g., link failure) in a timely manner remains a cornerstone of scalable networks. This means that SDN designs can rely upon a logical representation of centralized features (an abstraction layer that would support inter-PDP communications, for example).

4. Discussion

4.1. Implications of Full Automation

The path towards full automation is paved with numerous challenges and requirements, including:

- Making sure automation is well implemented so as to facilitate testing (including validation checks) and troubleshooting.
    - This suggests the need for simulation tools that accurately assess the impact of introducing a high level of automation in the overall service delivery procedure to avoid a typical "mad robot" syndrome, whose consequences can be serious from control and QoS standpoints, among others.
    - This also suggests careful management of human expertise, so that network operators can use robust, flexible means to automate repetitive or error-prone tasks and then build on automation or stringing together multiple actions to create increasingly complex tasks that require less human interaction (guidance and input) to complete.
- Simplifying and fostering service delivery, assurance, and fulfillment, as well as network failure detection, diagnosis, and root cause analysis for cost optimization.
    - Such cost optimization relates to improved service delivery times as well as optimized human expertise (see above) and global, technology-agnostic service structuring and delivery procedures. In particular, the ability to inject new functions in existing devices should not assume a replacement of the said devices but rather allow smart investment capitalization.
    - This can be achieved thanks to automation, possibly based upon a logically centralized view of the network infrastructure (or a portion thereof), yielding the need for highly automated topology, device and capabilities discovery means, and operational procedures.
    - The main intelligence resides in the PDP, which suggests that an important part of the SDN-related development effort should focus on a detailed specification of the PDP function, including algorithms and behavioral state machineries that are based upon a complete set of standardized data and information models.

  - These information models and data need to be carefully structured for efficiency and flexibility. This probably suggests that a set of simplified pseudo-blocks can be assembled as per the nature of the service to be delivered.

- The need for abstraction layers -- clear interfaces between business actors and between layers, let alone cross-layer considerations, etc. Such abstraction layers are invoked within the context of service structuring and packaging and are meant to facilitate the emergence of the following:

  - IP connectivity service exposure to customers, peers, applications, content/service providers, etc. (an example of this can be seen in [CPP]).

  - Solutions that accommodate IP connectivity service requirements with network engineering objectives.

  - Dynamically adaptive decision-making processes, which can properly operate according to a set of input data and metrics, such as current resource usage and demand, traffic forecasts and matrices, etc., all for the sake of highly responsive dynamic resource allocation and policy enforcement schemes.

- Better accommodation of technologically heterogeneous networking environments through the following:

  - Vendor-independent configuration procedures based upon the enforcement of vendor-agnostic generic policies instead of vendor-specific languages.

  - Tools to aid manageability and orchestrate resources.

  - Avoiding proxies and privileging direct interaction with engines (e.g., routing and forwarding).

4.2. Bootstrapping an SDN

Means to dynamically discover the functional capabilities of the devices that will be steered by a PDP intelligence for automated network service delivery need to be provided. This is because the acquisition of the information related to what the network is actually capable of will help structure the PDP intelligence so that policy provisioning information can be derived accordingly.

A typical example would consist in documenting a traffic engineering policy based upon the dynamic discovery of the various functions supported by the network devices, as a function of the services to be delivered, thus yielding the establishment of different routes towards the same destination depending on the nature of the traffic, the location of the functions that need to be invoked to forward such traffic, etc.

Such dynamic discovery capability can rely upon the exchange of specific information by means of an IGP or BGP between network devices or between network devices and the PDP in legacy networking environments. The PDP can also send unsolicited commands towards network devices to acquire the description of their functional capabilities in return and derive network and service topologies accordingly.

Of course, SDN techniques (as introduced in Section 2.4) could be deployed in an IGP-/BGP-free networking environment, but the SDN bootstrapping procedure in such an environment still assumes the support of the following capabilities:

- Dynamically discover SDN participating nodes (including the PDP) and their respective capabilities in a resilient manner, assuming the mutual authentication of the PDP and the participating devices Section 5. The integrity of the information exchanged between the PDP and the participating devices during the discovery phase must also be preserved;

- Dynamically connect the PDP to the participating nodes and avoid any forwarding loops;

- Dynamically enable network services as a function of the device capabilities and (possibly) what has been dynamically negotiated between the customer and the service provider;

- Dynamically check connectivity between the PDP and the participating nodes and between participating nodes for the delivery of a given network service (or a set thereof);

- Dynamically assess the reachability scope as a function of the service to be delivered;

- Dynamically detect and diagnose failures, and proceed with corrective actions accordingly.

Likewise, the means to dynamically acquire the descriptive information (including the base configuration) of any network device that may participate in the delivery of a given service should be provided so as to help the PDP structure the services that can be delivered as a function of the available resources, their location, etc.

In IGP-/BGP-free networking environments, a specific bootstrap protocol may thus be required to support the aforementioned capabilities for proper PDP- and SDN-capable device operation, in addition to the possible need for a specific additional network that would provide discovery and connectivity features.

In particular, SDN design and operation in IGP-/BGP-free environments should provide performances similar to those of legacy environments that run an IGP and BGP. For example, the underlying network should remain operational even if connection with the PDP has been lost. Furthermore, operators should assess the cost of introducing a new, specific bootstrap protocol compared to the cost of integrating the aforementioned capabilities in existing IGP/BGP protocol machineries.

Since SDN-related features can be grafted into an existing network infrastructure, they may not be all enabled at once from a bootstrapping perspective; a gradual approach can be adopted instead.

A typical deployment example would be to use an SDN decision-making process as an emulation platform that would help service providers and operators make appropriate technical choices before their actual deployment in the network.

Finally, the completion of the discovery procedure does not necessarily mean that the network is now fully operational. The operationality of the network usually assumes a robust design based upon resilience and high availability features.

4.3. Operating an SDN

From an Operations and Management (OAM) standpoint [RFC6291], running an SDN-capable network raises several issues such as those listed below:

- How do SDN service and network management blocks interact? For example, how the results of the dynamic negotiation of service parameters with a customer or a set thereof over a given period of time will affect the PDP decision-making process (resource allocation, path computation, etc.).

- What should be the appropriate OAM tools for SDN network operation (e.g., to check PDP or PEP reachability)?

- How can performance (expressed in terms of service delivery time, for example) be optimized when the activation of software modules is controlled by an external entity (typically a PDP)?

- To what extent does an SDN implementation ease network manageability, including service and network diagnosis?

- Should the "control and data plane separation" principle be applied to the whole network or a portion thereof, as a function of the nature of the services to be delivered or by taking into account the technology that is currently deployed?

- What is the impact on the service provider’s testing procedures and methodologies (that are used during validation and pre-deployment phases)? Particularly, (1) how test cases will be defined and executed when the activation of customized modules is supported, (2) what the methodology is to assess the behavior of SDN-controlled devices, (3) how test regression will be conducted, (4) etc.

- How do SDN techniques impact service fulfillment and assurance? How the resulting behavior of SDN devices (completion of configuration tasks, for example) should be assessed against what has been dynamically negotiated with a customer. How to measure the efficiency of dynamically enforced policies as a function of the service that has been delivered. How to measure that what has been delivered is compliant with what has been negotiated. What the impact is of SDN techniques on troubleshooting practice.

- Is there any risk to operate frozen architectures because of potential interoperability issues between a controlled device and an SDN controller?

- How does the introduction of SDN techniques affect the lifetime of legacy systems? Is there any risk of (rapidly) obsoleting existing technologies because of their hardware or software limitations?

The answers to the above questions are very likely to be service provider specific, depending on their technological and business environments.

4.4. The Intelligence Resides in the PDP

The proposed SDN definition in Section 2.3 assumes an intelligence that may reside in the control or the management planes (or both). This intelligence is typically represented by a Policy Decision Point (PDP) [RFC2753], which is one of the key functional components of the policy-based management framework.

SDN networking, therefore, relies upon PDP functions that are capable of processing various input data (traffic forecasts, outcomes of negotiation between customers and service providers, resource status as depicted in appropriate information models instantiated in the PIB, etc.) to make appropriate decisions.

The design and the operation of such PDP-based intelligence in a scalable manner remains a part of the major areas that need to be investigated.

To avoid centralized design schemes, inter-PDP communication is likely to be required, and corresponding issues and solutions should be considered. Several PDP instances may thus be activated in a given domain. Because each of these PDP instances may be responsible for making decisions about the enforcement of a specific policy (e.g., one PDP for QoS policy enforcement purposes, another one for security policy enforcement purposes, etc.), an inter-PDP communication scheme is required for global PDP coordination and correlation.

Inter-domain PDP exchanges may also be needed for specific usages. Examples of such exchanges are as follows: (1) during the network attachment phase of a node to a visited network, the PDP operated by the visited network can contact the home PDP to retrieve the policies to be enforced for that node, and (2) various PDPs can collaborate in order to compute inter-domain paths that satisfy a set of traffic performance guarantees.

4.5. Simplicity and Adaptability vs. Complexity

The functional metadomains introduced in Section 2.4 assume the introduction of a high level of automation, from service negotiation to delivery and operation. Automation is the key to simplicity, but it must not be seen as a magic button that would be hit by a network administrator whenever a customer request has to be processed or additional resources need to be allocated.

The need for simplicity and adaptability, thanks to automated procedures, generally assumes some complexity that lies beneath automation.

4.6. Performance and Scalability

The combination of flexibility with software inevitably raises performance and scalability issues as a function of the number and the nature of the services to be delivered and their associated dynamics.

For example, networks deployed in Data Centers (DCs) and that rely upon OpenFlow switches are unlikely to raise important FIB scalability issues. Conversely, DC interconnect designs that aim to dynamically manage Virtual Machine (VM) mobility, possibly based upon the dynamic enforcement of specific QoS policies, may raise scalability issues.

The claimed flexibility of SDN networking in the latter context will have to be carefully investigated by operators.

4.7. Risk Assessment

Various risks are to be assessed such as:

- Evaluating the risk of depending on a controller technology rather than a device technology.

- Evaluating the risk of operating frozen architectures because of potential interoperability issues between a controller and a controlled device.

- Assessing whether SDN-labeled solutions are likely to obsolete existing technologies because of hardware limitations. From a technical standpoint, the ability to dynamically provision resources as a function of the services to be delivered may be incompatible with legacy routing systems because of their hardware limitations, for example. Likewise, from an economical standpoint, the use of SDN solutions for the sake of flexibility and automation may dramatically impact Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) budgets.

5. Security Considerations

Security is an important aspect of any SDN design because it conditions the robustness and reliability of the interactions between network and applications people for efficient access control procedures and optimized protection of SDN resources against any kind of attack. In particular, SDN security policies [SDNSEC] should make sure that SDN resources are properly safeguarded against actions that may jeopardize network or application operations.

In particular, service providers should define procedures to assess the reliability of software modules embedded in SDN nodes. Such procedures should include the means to also assess the behavior of software components (under stress conditions), detect any exploitable vulnerability, reliably proceed with software upgrades, etc. These security guards should be activated during initial SDN node deployment and activation but also during SDN operation that implies software upgrade procedures.

Although these procedures may not be SDN-specific (e.g., operators are familiar with firmware updates with or without service disruption), it is worth challenging existing practice in light of SDN deployment and operation.

Likewise, PEP-PDP interactions suggest the need to make sure that (1) a PDP is entitled to solicit PEPs, so that they can apply the decisions made by the said PDP, (2) a PEP is entitled to solicit a PDP for whatever reason (request for additional configuration information, notification about the results of a set of configuration tasks, etc.), (3) a PEP can accept decisions made by a PDP, and (4) communication between PDPs within a domain or between domains is properly secured (e.g., make sure a pair of PDPs are entitled to communicate with each other, make sure the confidentiality of the information exchanged between two PDPs can be preserved, etc.).

6. Acknowledgements

Many thanks to R. Barnes, S. Bryant, S. Dawkins, A. Farrel, S. Farrell, W. George, J. Halpern, D. King, J. Hadi Salim, and T. Tsou for their comments. Special thanks to P. Georgatos for the fruitful discussions on SDN Interconnection (SDNI) in particular.

7. Informative References

[AN] Tennenhouse, D. and D. Wetherall, "Towards an Active Network Architecture", Multimedia Computing and Networking (MMCN), January 1996.

[AUTOMATION] Boucadair, M. and C. Jacquenet, "Requirements for Automated (Configuration) Management", Work in Progress, January 2014.

[CPNP] Boucadair, M. and C. Jacquenet, "Connectivity Provisioning Negotiation Protocol (CPNP)", Work in Progress, October 2013.

[CPP] Boucadair, M., Jacquenet, C., and N. Wang, "IP/MPLS Connectivity Provisioning Profile", Work in Progress, September 2012.

[LS-DISTRIB] Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", Work in Progress, November 2013.

[PN] Campbell, A., De Meer, H., Kounavis, M., Kazuho, M., Vincente, J., and D. Villela, "A Survey of Programmable Networks", ACM SIGCOMM Computer Communication Review, April 1999.

[RFC1383] Huitema, C., "An Experiment in DNS Based IP Routing", RFC 1383, December 1992.

[RFC2622] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D., and M. Terpstra, "Routing Policy Specification Language (RPSL)", RFC 2622, June 1999.

[RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

[RFC3084] Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie, K., Herzog, S., Reichmeyer, F., Yavatkar, R., and A. Smith, "COPS Usage for Policy Provisioning (COPS-PR)", RFC 3084, March 2001.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC5440] Vasseur, JP. and JL. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, March 2009.

[RFC5810] Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W., Dong, L., Gopal, R., and J. Halpern, "Forwarding and Control Element Separation (ForCES) Protocol Specification", RFC 5810, March 2010.

[RFC5812] Halpern, J. and J. Hadi Salim, "Forwarding and Control Element Separation (ForCES) Forwarding Element Model", RFC 5812, March 2010.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, June 2011.

[SDNSEC] Hartman, S. and D. Zhang, "Security Requirements in the Software Defined Networking Model", Work in Progress, April 2013.

[SLA-EXCHANGE] Shah, S., Patel, K., Bajaj, S., Tomotaki, L., and M. Boucadair, "Inter-domain SLA Exchange", Work in Progress, November 2013.

Authors’ Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet
France Telecom
Rennes
France

EMail: christian.jacquenet@orange.com