SDNRG Working Group
Internet-Draft
Intended status: Informational
Expires: October 12, 2013
M. Boucadair
C. Jacquenet
France Telecom
April 10, 2013

Software-Defined Networking: A Service Provider's Perspective
draft-sin-sdnrg-sdn-approach-02

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 at contributing to the clarification of the SDN landscape.

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 a 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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 12, 2013.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

1.1. Context

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 capabilities (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

capabilities (e.g., traffic classification and marking, traffic conditioning and scheduling), security capabilities (e.g., filters to protect customer premises from network-originated attacks, to avoid malformed route announcements, etc.) and management capabilities (e.g., detection and processing of faults).

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 Any Time Any-Where Any Device (ATAWAD)-driven convergence trends that are meant to make sure 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 this end-user is in motion or not.

Yet, most of these services have been deployed for the past decade, based solely on often static service production procedures that are more and more exposed to the risk of malformed 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 the market coverage (fixed/mobile, residential/corporate). This so-called network planning policy may very well affect the way resources are allocated in a network, but clearly fails to be adequately responsive to highly dynamic customer requirements in an "always-on" fashion. These needs are 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 the sake of event aggregation, correlation and processing. At the cost of extra complexity and possible customer's 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.

1.2. Scope

This document is a contribution to clarify the SDN landscape:

- o Section 2 clarifies items which are considered as viable goals and exclude others from the scope.
- o Section 3 provides a tentative definition from a service provider perspective.
- o Section 3 discusses several issues and identifies some requirements.

2. What is In and What is Out?

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 against 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.

2.1. Remember the Past

SDN techniques cannot be seen as a brand new solution but rather as some kind of rebranding of proposals that have been investigated for several years, like Active or Programmable Networks. As a matter of fact, some of the claimed "new" SDN features have been already implemented (e.g., NMS (Network Management System), PCE (Path Computation Element, [RFC4655])), and supported by vendors for quite some time (references can be added if needed).

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 ForCES ([RFC5810][RFC5812])).

2.2. Be Pragmatic

SDN approaches should be holistic. This means that it must be global, network-wise. It is not a matter of configuring devices one by one to enforce a specific forwarding policy. It is about configuring and operating a whole range of devices at the scale of the network for the sake of automated service delivery ([I-D.boucadair-network-automation-requirements]), from service negotiation and creation (e.g., [I-D.ietf-idr-sla-exchange]) to assurance and fulfillment.

Because the complexity of activating SDN capabilities is hidden (to the user) and pushed to software, 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 (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 techniques remain limited to what is supported by embedded software and hardware. One cannot expect SDN techniques to support unlimited customizable features.

Policy-based management framework[RFC2753] 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).

SDN techniques as a whole are an instantiation of the policy-based network 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) and triggers (e.g., dynamic notification of a link failure), etc.

2.3. Measure Experience Against Expectations

Because several software modules may be controlled by external entities, means to ensure the experienced outcome complies with the expected outcome belong to the set of SDN techniques.

These techniques, as an instantiation of Policy-Based Management, should interact with Service Structuring engines and the network to continuously assess whether the experienced network behavior is compliant with the objectives set by the Service Structuring engine, and which may have been dynamically negotiated with the customer (e.g., captured in a CPP

[I-D.boucadair-connectivity-provisioning-profile]). This requirement applies to several regions of a network:

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 and ordering, through order processing, to delivery, assurance and fulfillment, should be supported. This approach assumes widely adopted standard data and information models, let alone interfaces.

2.4. Design Carefully

Exposing open and programmable interfaces has a cost, from both a 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, forwarding) without requiring any in-between adaptation layer (generic objects to vendor-specific CLI commands for instance).

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-wise deployment is unlikely.

Instead, multiple instantiations of SDN techniques will be progressively deployed and adapted to various network and service segments.

2.5. There is Life Beyond OpenFlow

Empowering networking with in-band controllable modules does not necessarily mean the use of the OpenFlow protocol, which is only a protocol that helps devices populate their forwarding tables according to a set of instructions.

OpenFlow is clearly not the "next big thing": there are many, many other protocols that have been standardized (think Routing Policy Specification Language (RPSL, [RFC2622]), for one) - or not - and which have been massively deployed.

The forwarding of the configuration information can currently rely upon a variety of protocols that include (but is not necessarily limited to) PCEP [RFC5440], NETCONF [RFC6241], COPS-PR [RFC3084], etc.

There is 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.

2.6. Non Goals

There are inevitable trade-offs between the current networking ecosystem and the proposed SDN paradigm. Operators do not have to choose between the two as both models may be needed.

In particular, the following considerations can be seen as a non-goal to justify the deployment of SDN techniques:

- Fully flexible software implementations, whereas the claimed flexibility will be limited by respective 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 raise some scalability issues. Distributed protocols and their ability to react to some events (e.g., link failure) in a timely manner remains a key to 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).

3. A Definition of Software-Defined Networking

3.1. A Tautology

The separation of the forwarding and control planes (beyond implementation considerations) have almost become a gimmick to promote flexibility as a key feature of the SDN approach. Technically, most of current router implementations have been assuming this separation for years if not decades. Routing processes (such as IGP and BGP route computation) have often been software-based, while forwarding capabilities are hardware-encoded.

As such, the current 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 route computation before FIB population is certainly another story.

3.2. On Flexibility

This "flexibility argument" that has been put forward by SDN promoters 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's requests for the sake of 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. Note:

- o The exact characterization of what flexibility actually means is still required.
- o The exposure of programmable interfaces is not a goal per se, rather a means to facilitate configuration procedures.

3.3. A Tentative Definition

We define Software-Defined Networking as the set of techniques used to facilitate the design, the delivery and the operation of network services in a deterministic, dynamic, and scalable manner.

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

Because networking is by essence software-driven, the above definition does not emphasize the claimed "Softwire-Defined" property of SDN-labeled solutions.

Having a predictable network behavior is important to consider. This argues in favor of investigating advanced network emulation engines which would assess the impact of enforcing a new policy. Network emulation function would be a helper for operators. A network emulation engine can be fed with information collected using for instance [I-D.ietf-idr-ls-distribution].

3.4. Functional Meta-Domains

SDN techniques can be classified into the following functional meta-domains:

- o Techniques for the dynamic discovery of network topology, devices and capabilities, along with relevant information models that are meant to precisely document such topology, devices and capabilities.
- o Techniques for exposing network services (and their characteristics; e.g., [I-D.boucadair-connectivity-provisioning-profile]) and for dynamic negotiation of the set of corresponding parameters that will be used to measure the level of quality associated to the delivery of a given service or a combination thereof.
- o Techniques used by service requirements-derived dynamic resource allocation and policy enforcement schemes, so that networks can be programmed accordingly.
- o 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.

4. Disscussion

4.1. Full Automation: a Viable Objective?

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

- o Simplify and foster service delivery, assurance and fulfillment, as well as network failure detection, diagnosis and root cause analysis:

  * 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 as well as operational procedures.
  * The main intelligence resides in the PDP, which suggests that an important part of the investigation effort should focus on a detailed specification of the PDP function, including algorithms and behavioral details, based upon a complete set of standardized data and information models.
- o Need for abstraction layers: clear interfaces between business actors, clear interaction between layers, cross-layer considerations, etc.

  * Ability to build and package differentiated (network) services.

   * Need for IP connectivity service exposure to customers, peers, applications, content/service providers, etc. (e.g., [I-D.boucadair-connectivity-provisioning-profile]).
   * Need for a solution to map IP connectivity service requirements with network engineering objectives.
   * Need for dynamically-adaptive objectives based on current resource usage and demand, for the sake of highly responsive dynamic resource allocation and policy enforcement schemes.
- o Better accommodate technologically heterogeneous networking environments:

   * Need for vendor-independent configuration procedures, based upon the enforcement of vendor-agnostic generic policies instead of vendor-specific languages.
   * Need for tools to aid manageability and orchestrate resources.
   * Avoid proxies and privileged direct interaction with engines (e.g., routing, forwarding).

4.2. The Intelligence resides in the PDP

The proposed SDN definition in Section 3.3 assumes an intelligence that may reside in the control or management planes (or both). This intelligence is typically represented by a Policy Decision Point, which is one of the key functional components of Policy-Based Management architectures [RFC2753].

The Policy Decision Point (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.

The Policy Enforcement Point (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 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 of the major areas that needs to be investigated within SDN environments.

To avoid centralized design schemes, inter-PDP communication means should be considered.

Several PDP instances may be activated in a given domain. Because each of these PDP instances may be in charge of a given functional perimeter, an inter-PDP communication may be required to ease collaboration between these PDPs to provision a network service.

Inter-domain PDP exchanges may be needed for some specific usages. Examples of such exchanges are: (1) During the network attachment phase of a node to a visited network, the PDP belonging to the the visited network can contact the home PDP to retrieve the policies to be enforced for that node. (2) Various PDPs can collaborate together in order to compute inter-domain paths which satisfy a set of traffic performance guarantees.

4.3. **Simplicity and Adaptability vs.** Complexity

The above meta functional domains assume the introduction of a high level of automation, from service negotiation to delivery and operation.

Automation is the key to simplicity, but 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.

4.4. Performance & 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.

While the deployment of a network solely composed of OpenFlow switches within a data center environment is unlikely to raise FIB scalability issues given the current state-of-the-art, data center networking that relies upon complex, possibly IP-based, QoS-inferred, interconnect design schemes meant to dynamically manage the mobility of Virtual Machines between sites is certainly another scale.

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

4.5. Risk Assessement

Various risks are to be assessed such as:

- o Evaluating the risk of depending on a controller technology rather than a device technology.
- o Evaluating the risk of operating frozen architectures because of potential interoperability issues between a controller and a controlled device.
- o Assessing whether SDN-labeled solutions are likely to obsolete existing technologies because of hardware limitations.
- o Etc.

5. IANA Considerations

This document does not require any action from IANA.

6. Security Considerations

This document does not define any protocol nor architecture.

7. Acknowledgements

Many thanks to J. Halpern and T. Tsou for their feedback.

Special thanks to P. Georgatos for the interesting discussion; particularly the discussion on SDNi (SDN Interconnection).

8. Informative References

[I-D.boucadair-connectivity-provisioning-profile]
Boucadair, M., Jacquenet, C., and N. Wang, "IP/MPLS Connectivity Provisioning Profile", draft-boucadair-connectivity-provisioning-profile-02 (work in progress), September 2012.

[I-D.boucadair-network-automation-requirements]
Boucadair, M. and C. Jacquenet, "Requirements for Automated (Configuration) Management", draft-boucadair-network-automation-requirements-00 (work in progress), December 2012.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE

Information using BGP", draft-ietf-idr-ls-distribution-02 (work in progress), February 2013.

[I-D.ietf-idr-sla-exchange]
Shah, S., Patel, K., Bajaj, S., Tomotaki, L., and M. Boucadair, "Inter-domain SLA Exchange", draft-ietf-idr-sla-exchange-00 (work in progress), January 2013.

[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.-P., 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.

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