Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 17, 2015

M. Boucadair
C. Jacquenet
France Telecom
L. Contreras
Telefonica I+D

February 13, 2015

Requirements for Automated (Configuration) Management
draft-boucadair-network-automation-requirements-05

Abstract

Given the ever-increasing complexity of the configuration tasks required for the dynamic provisioning of IP networks and services, this document aims at listing the requirements for an automated configuration management framework.

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 August 17, 2015.

Table of Contents

1. Introduction

IP network and service configuration procedures are currently handled by skilled personnel who is often required to acquire a high level of expertise that grows as the variety and the complexity of the

services to be delivered over an IP network. This demand for a high level of expertise is further increased by heterogeneous network and service environments where each equipment manufacturer has developed its own proprietary interfaces and configuration schemes. As a consequence, the time to deliver complex yet advanced IP service offerings (such as IP TV, VPN, etc.) is also increasing at the risk of jeopardizing customers' quality of experience.

This document advocates for the need to undertake a standardization effort to define an automated provisioning framework that includes a set of interfaces and protocol(s) for conveying configuration information which should help in facilitating the automation of the network resource allocation and service delivery procedures. Defining standard data and information models [RFC3444] to capture offered network services would help to automate the process of service ordering and activation and therefore accelerating service provisioning.

Automation should not be targeted at dynamically enforcing policies only, but also be encouraged to:

- Generate policy-related and configuration data based on a well-defined set of triggers and events.
- Monitor the outcome of a configured function/device to assess whether the observed behavior is aligned with the expected behavior.

This document assumes that service differentiation at the network layer can be enforced by tweaking various parameters which belong to distinct dimensions (e.g, forwarding, routing, traffic access management, traffic classification, etc.). As such, the decision point is likely to interact with several engines (e.g., routing engine, forwarding engine, etc.). In particular, this document considers that an I2RS system can be seen as a subset of an overall framework. I2RS is limited to routing and forwarding actions (see Section 7.2.2). To meet performance requirements (see Section 7.1.2), it is encouraged to design a system which interacts directly with the routing and forwarding system, rather than requiring local proxy functions which are responsible for translating vendor-independent commands and policies into vendor-specific configuration commands and syntax.

In addition to protocol-related considerations, automating network operations heavily relies upon the availability of intelligent policy decision points. Sharing best design practices for policy decision point logics would facilitate the adoption of the proposed approach (see Section 6).

The document enumerates a set of encountered issues (see Section 5) and identifies a set of requirements (see Section 7). A service-driven approach is purposed in Section 6.

2. Terminology

This document makes use of the following terms:

- Decision point: is an entity that is responsible for making decisions that yield the production of configuration information which will be conveyed towards (and processed by) the set of relevant managed entities.
- Managed entity: any (networking) device that will participate in the establishment, the activation and the maintenance of a given service. Such devices MAY include routers and terminals, whatever the configuration procedures and underlying technologies to be used for the delivery of the said service.

3. Scope & Overall Context

Maintain and operate self-adaptive networks may be seen as a long term objective for IP service providers. To achieve this goal, intermediate objectives should be defined, such as:

1. Define a framework to expose IP connectivity services to external parties, including peering IP network operators, content providers, services relying on connectivity services (e.g., IP TV, VoIP) (see for example [RFC7297]).
2. Ability to automatically translate IP connectivity requirements into configuration and provision actions.
3. Dynamically adapt service configuration to be aligned with expected service objectives.
4. Automate service negotiation and service activation (e.g., [I-D.boucadair-connectivity-provisioning-protocol]).
5. Optimize resource utilization, e.g., automatically set traffic engineering objectives.

Discussing the items above is out of scope. This document only discusses requirements for (automated) configuration procedures and protocol.

4. Motivations

Service providers and network operators have gained experience in implementing, deploying and manipulating a large set of protocols and associated information. Some data models have also been defined for network management purposes. Thus, several protocols have been standardized, such as SNMP (Simple Network Management Protocol

[RFC3410]), COPS (Common Open Policy Service [RFC2748]), (COPS-PR [RFC3084]) or, more recently, NETCONF [RFC6241].

In addition, multiple data models have been defined and used by operators like CIM (Core Information Model), DEN (Directory-Enabled Network), SMI (Structure of Management Information [RFC2578]), SPPI (Structure of Policy Provisioning Information [RFC3159]), and, more recently, YANG [RFC6022].

Despite this standardization effort, most of the service operators still assume manual configuration through proprietary CLI (Command Line Interface) commands possibly combined with in-house developed, vendor-specific scripts to proceed with the configuration of numerous features, such as forwarding and routing capabilities, Quality of Service (sometimes including traffic engineering) capabilities, and security capabilities. Some of these requirements are fulfilled by existing tools/protocols but there is still a lack of wide adoption of those tools.

Other non-technological challenges are also to be taken into consideration when discussing network automation (e.g., to what extent an automated system will accommodate both simple and complex business scenarios, how an automated system will evolve to accommodate changes and new procedures, assess the impact on testing methodologies,etc.).

The purpose of this document is to document requirements rather than focusing on the non-technological challenges.

5. Issues Raised by Configuration Operations

The following sub-sections enumerates a set of issues.

5.1. Heterogeneous Environments

The delivery of IP services relies upon the activation of a set of capabilities located in various devices that include routers, switches, service platforms, etc. In particular, a large set of protocols need to be configured, such as routing protocols, management protocols, security protocols, let alone capabilities that relate to addressing scheme management, policy enforcement, etc.

Such a diversity of features and protocols may increase the risk of inconsistency at the cost of QoS degradation or even service disruption. Therefore, the configuration information which is forwarded to the whole set of participating devices for delivering a given service or a set of services should be consistent, whatever the number of features/services to be activated/deployed in the network.

5.2. Complex Topologies

Network operators should have means to dynamically discover the topology of the network.

Such topological information should be as elaborate as possible, including details like the links that connect network devices, their capacity, such as the total bandwidth, the available bandwidth, the bandwidth that can be reserved, etc.

5.3. Multi-Functional Devices

Numerous, often multi-vendor devices are involved in the delivery of IP services. These devices support various capabilities that need to be combined for the delivery of a given service or a set of services. The availability and status of such capabilities is therefore a critical information for service providers, since it is likely to affect service and network design, let alone operational procedures.

Therefore, service providers and network operators should have means to:

- Dynamically retrieve, list and classify the capabilities supported by a given device (or a set thereof),
- Dynamically acquire detailed information about the availability and status of any activated capability of any device at any given time.
- Dynamically retrieve the version of embedded software modules, interfaces, OS version, etc.

5.4. Performance Impacts

Configuring a set of devices to deliver a service takes time. In addition, depending on the complexity of the service, erroneous configurations may occur at the cost of jeopardizing the overall quality of a service, if not causing service disruption. From this perspective, some performance indicators must be defined and measured to assess:

- The time to deliver a service, from subscription to operation. Such indicator may be further decomposed into elementary performance metrics, e.g., the time it takes to complete the configurations tasks that are specific to the enforcement of a given policy (forwarding, routing, QoS, etc.)
- The impact of any configuration change on the overall service performance (including customer's own perception).

Tools to qualify (by simulation or emulation) any possible impact of an elementary configuration task before such task is performed should be supported. These tools aims to prevent errors amplification.

5.5. Scalability

As far as scalability is concerned, adequate indicators should be specified in order to assess the ability of configuration techniques and protocols to support a large number of simultaneous processes. The maintenance of these processes should not impact the performance of the configuration system as a whole (i.e., manager and managed entities, amount of configuration task-specific traffic exchanged between manager and managed entities, periodicity of configuration operations, etc.).

Therefore, configuration operations should be qualified with performance indicators in order to check whether the architecture designed for configuration management is scalable in terms of:

- Amount of configuration data to be processed per unit of time, as a function of the number and the nature of the capabilities and devices that need to be configured.
- Amount of traffic generated by any reporting mechanism that may be associated to a configuration process.
- Number of processes that are created in order to achieve specific configuration operations.

5.6. Limits of Manual Configuration

Manual configuration is not only a likely source of errors, but it also affects the time it takes to complete a configuration task (or a combination thereof) to deliver a service, as a function of the task complexity and the need for global consistency. Thus, the efficiency of a configuration process is likely to be improved by the introduction of a high level of automation. Automation is defined as follows:

- Automatic provisioning of configuration information to the participating devices.
- Dynamic enforcement of policies (possibly based upon the use of dynamic resource allocation techniques).
- Dynamic reporting mechanisms to notify about the actual processing of configuration information by a participating device.
- Autonomic provisioning capabilities for triggering self-configuration mechanisms for the network devices.

Refer to Section 4.1 of [RFC7149] for a discussion on the implications of full automation.

5.7. Security Issues

Configuring a network or a service raises several security issues, including (but not limited to):

- The integrity of the configuration information, possibly yielding the preservation of the confidentiality of such information when being forwarded over a public IP infrastructure,
- The need for authorizing and authenticating devices/entities that have the ability of manipulating configuration information (define, instantiate, forward and process),
- Mutual authentication between manager and managed entities.

6. Introducing Service-Driven Configuration Management

Current practice consists in configuring elementary functions, i.e., configuration management for a given service offering is decomposed into a set of elementary tasks. Thus, the consistency of configuration operations for the sake of service delivery must be checked by any means appropriate.

A network device should be seen as a means to deploy a service and not just as a component of such service. Thus, service delivery procedures should not assume the configuration of devices one after the other, but rather globally, i.e., at the scale of the network that supports the said service. Such a service-driven configuration management scheme is therefore meant to facilitate and improve the completion of configuration tasks, by means of highly automated, service-wise, global configuration procedures.

This in particular assumes the need for robust configuration mechanisms that include appropriate protocol machinery (e.g., from a reliable transport mode perspective) to convey configuration information between manager and managed entities, as well as reliable consistency check procedures. The latter is not only meant to assess the validity of all the configuration operations service-wise, but also the efficiency of the corresponding yet dynamic policy enforcement and resource allocation schemes.

An implementation example is the case of service providers who could dedicate (logical) centralized entities which are responsible for the provisioning and the management of participating devices. The main function of these centralized entities is to make appropriate decisions and generate the decision-derived configuration data that will be forwarded to the participating devices. In addition, these centralized entities will make sure of the consistency of the decisions that have been made to deliver the service, according to a dynamic configuration policy enforcement scheme. These logical

entities will be responsible for assessing whether the enforced policies are compliant with the expected behavior and how efficiently they are enforced.

Service-driven configuration management leads to the following assumptions:

- Data and information models must be service-oriented,
- Configuration protocol(s) should reuse existing standard data and information models as much as possible,
- Configuration protocol(s) should be flexible enough to facilitate the support of new features without compromising the protocol robustness (especially from a performance and scalability standpoints),
- Configuration protocol(s) should provide means to check the consistency of configuration information service-wise.

7. Detailed Requirements

7.1. Protocol Requirements

Configuration information must be provided to the participating devices by means of a protocol to be used between such devices and a presumably centralized manager entity. The latter can be seen as a decision point where configuration information is stored, maintained and updated whenever required.

Decisions about configuring additional features or devices, enforcing policies and allocating resources are made accordingly, e.g., as a function of the number of Service Level Specification templates that are processed per unit of time combined with traffic forecasts that are updated on a regular basis. Such decisions are converted into configuration information that is forwarded towards the relevant managed entities.

7.1.1. Functional Requirements

The vendor-independent communication protocol for conveying configuration information should have the following characteristics:

1. The protocol must be reliable, and be independent from the network layer (i.e., configuration information must be conveyed over IPv4 and IPv6 network infrastructures indifferently),
2. The protocol architecture should provide a means for dynamically providing the configuration information to the participating devices, so that a high level of automation is introduced in the actual delivery of any given service.

3. The protocol should provide the relevant means (encoding capabilities, operation and command primitives, extension capabilities that allow additional operations, etc.) to be able to reliably and securely convey configuration information,
4. The protocol should be a privileged vector for the dynamic provisioning of configuration data, as well as the dynamic enforcement of any policy such as a routing policy, a QoS policy or a security policy. This requirement suggests the definition and the support of vendor-independent instantiation procedures that will aim at uniquely identifying the configuration data model and the policy enforcement scheme that refer to a given IP service.
5. The protocol should support a reporting mechanism for various purposes, including the assessment of the efficiency of a given policy, the ability to dynamically notify the aforementioned decision point about the completion of a set of configuration tasks, or the ability to dynamically report any event that may affect global service operation,
6. The protocol should support the appropriate security mechanisms to provide guarantees as far as the preservation of the confidentiality of the configuration information is concerned.
7. The protocol should provide a mean of preserving the order in which the configuration information should be applied in the participating devices. The ordering of the configuration information could be implemented by means of sequence numbers, timing or scheduling indicators, etc. Through this requirement, any aged or disordered configuration information is prevented to be applied to the devices.

7.1.2. Performance Requirements

The protocol for conveying configuration information within a network should be designed so that:

1. The activation of the protocol by the participating devices must not affect the overall performance of such devices, whatever the amount of configuration data these devices will have to process at any given time.
2. The activation of the protocol should not dramatically affect the global resources of the network infrastructure that will convey configuration information whatever its amount and scope (e.g., the set of policies that need to be dynamically enforced).

7.1.3. Backward Compatibility

The introduction and the activation of a protocol for conveying configuration information should allow for smooth migration procedures, so that vendor-specific and vendor-independent

configuration procedures may gracefully co-exist on a (hopefully) limited period of time.

Also, in case of any kind of protocol failure, it must be possible to rely upon any vendor-specific configuration procedure as some kind of rollback procedure. Such a rollback procedure must protect services that are up and running from any risk of disruption.

7.2. Requirements for Configuration Information

Configuration tasks are currently performed with vendor-specific solutions that reflect technology-specific information. It is therefore more and more difficult for a service provider to get a unified, homogeneous view of the network resources service-wise (rather than device-wise).

Configuration information should therefore be provided to the participating devices as unified, vendor-agnostic, service configuration parameters. These parameters must reflect a standardized service data model rather than a vendor-specific information model, unlike the current situation. Examples of such service data models include a tunneling service, an intra-domain routing service, or a VPN service.

The need for a unified, homogeneous access to a multi-vendor environment is becoming critical for N-Play, residential and corporate, fixed and mobile service providers so that a high level of automation can be introduced while proceeding with the configuration of the said multi-vendor environment. This unification is clearly conditioned by the availability of two key components: A configuration protocol (the container) and a set of data models (the content).

The standardization of these two components has several yet major benefits:

- Devices are seen as functional blocks that support a set of standardized capabilities;
- These functional blocks are described as vendor-independent capabilities;
- These functional blocks are all managed homogeneously, whatever the underlying technology.

As a consequence, it becomes possible to add semantic rules to automate detection and correction of erroneous configurations, either at the scale of a single device or at the scale of a whole network. Furthermore, an equipment from vendor X could de replaced by another

technology from vendor Y with very little impact (if no impact at all) on the configuration management procedures.

To do so, the data models should satisfy the following requirements.

7.2.1. Network Services

7.2.1.1. Interface Identification

Configuration information for identification purposes mostly deals with the naming of any interface supported by a given device. This naming scheme describes the properties of an interface, and must take into account all the parameters that are required to correctly configure an interface. The following information must be provided:

- A name, with a generic syntax that is vendor-agnostic by nature. The name can define the media type of the interface. Depending on the medium type, further information MAY be added (such as MTU, bandwidth, supported framing and encapsulation modes, etc.).
- The interface technology (e.g., optical / electrical) and nominal capacity (e.g., 10 GE / 100 GE).
- Optionally, a logical descriptor (e.g., VLANs declared on Ethernet interfaces). In this case the encapsulation mode must be part of the configuration information.
- Optionally, a description field that provides general (possibly administrative) information about the interface.

7.2.1.2. Quality of Service (QoS)

IP services are provided with a level of quality that MAY be guaranteed (either qualitatively or quantitatively) by any means appropriate. QoS policies should be dynamically enforced according to a data model that will accurately reflect all the elementary QoS capabilities that MAY be configured and activated to enforce such policies.

For instance, in the case of the activation of the Diffserv QoS model within a network infrastructure, the participating routers should be provided with the appropriate PHB (Per Hop Behavior) configuration parameters.

Additional information relevant to the service, such as path protection, can be provided to the participating devices to mitigate network failures. This information can be proactively or reactively provided, according to the service level agreed.

7.2.1.3. Applications

Network devices usually support functions that allow the activation of specific services like HTTP, BOOTP, DHCP, SYSLOG, etc. These devices must therefore be provided with the corresponding configuration information.

7.2.2. Forwarding Services

7.2.2.1. Routing and Forwarding Configuration Information

Routing and forwarding configuration information deals with the decision that should be applied by a participating device to forward an incoming IP datagram, according to the (possibly service-specific) forwarding and routing policies defined by the service provider. From this perspective, the participating devices should be provided with the following configuration information:

1. Metric information for IGP route computation purposes,
2. Attribute information for BGP route computation purposes,
3. Static routes (if any).

Any candidate protocol must be compliant with the following requirements:

1. Ability to retrieve routing and forwarding tables.
2. Ability to retrieve the configuration information of each routing/forwarding device.
3. Ability to retrieve the capabilities of each routing/forwarding device.
4. Ability to dynamically enforce policies on active routing processes.
5. Ability to dynamically inject new routing and forwarding entries.
6. Ability to receive notifications when route changes occurred, tagged by the decision point.

7.2.2.2. Traffic Engineering Configuration Information

Traffic Engineering (TE) is an important and often complex task of configuration management: the participating devices should be provided with the configuration information that will help them to select the appropriate routes that lead to a set of destinations, according to specific constraints and requirements that may have been dynamically negotiated with the customer.

These constraints may be expressed in terms of time duration (e.g., the use of a traffic-engineered route on a weekly basis), traffic characterization (e.g., all IP multicast traffic should be forwarded

along a specific distribution tree), security concerns (e.g., use IPsec tunnels), and/or QoS considerations (e.g., EF (Expedited Forwarding)-marked traffic [RFC3246] should always use a subset of "EF-compliant" routes).

7.2.2.3. Configuration Information for Tunnel Design and Activation

7.2.2.3.1. Tunnel Identification Information

The identification of a tunnel should be globally unique, especially if the tunnel is to be established and activated across autonomous systems. The tunnel identification schemes (e.g., endpoint numbering) should be left to service providers, assuming that the corresponding formalism is commonly understood, whatever the number of autonomous systems the tunnel may cross.

The tunnel identification information should at least be composed of the tunnel endpoint identification information. The tunnel identification information MAY also be composed of an informal description of the tunnel, e.g., the purpose of its establishment, customer traffic that may be forwarded into this tunnel, etc.

7.2.2.3.2. Tunneling Protocol Configuration Information

Any participating device must be provided with the configuration information related to the tunneling technique to be used for the establishment and the activation of the tunnel. Such techniques include Generic Routing Encapsulation (GRE, [RFC2784]), IP Secure in tunnel mode (IPsec, [RFC2401]), Layer 2 Tunneling Protocol (L2TP, [RFC2661]), etc.

7.3. Global Management Requirements

7.3.1. Fault Management

Mechanisms to monitor and report any fault that affects service operation should be independent of the configuration protocol.

7.3.2. Configuration Management

Errors during a configuration procedure could impact the availability of a given service offering, while consistency checks are mandatory so as to correctly enforce a configuration policy.

The following requirements have been identified:

- Provisioning of configuration information should be as automated as possible,

- Mechanisms to detect and diagnose configuration errors must be supported,
- Consistency of configuration operations service-wise must be checked,
- Simulation tools should be used to assess the validity of configuration information before it is downloaded to the relevant participant devices.
- Autonomic provisioning capabilities should be enabled to facilitate new device deployments in an automatic way, ideally without any human configuration intervention. Of course, the procedure must be designed to allow for administrative validation under some events. The purpose of allowing for such events is to ease troubleshooting and react to failures events when unexpected behaviors are experienced.
- Means to prevent against "mad robot" phenomena should be supported.

7.3.3. Performance Management

Performance management is key for guaranteeing Service Assurance by proactively detecting network degradation.

In a vendor-agnostic scenario, the mechanisms for performance management should implement standardized measurements among the involved devices, represented by abstract, standard data models. There are a number of measurements that can be taken into account for different purposes, such as CPP validation, bandwidth utilization or network and service level resilience. To that end, the performance management tools should provide reporting capabilities of the obtained measurements through counters or any other mean agnostic to specific vendor implementations.

The activation (and de-activation) of the reporting capabilities MAY be enabled by using automated configuration mechanisms.

7.4. Security Management

7.4.1. Device Authentication

It must be possible to activate mutual authentication between manager and managed entities. The authentication must be checked before exchanging any configuration data, so as to prevent DoS (Denial of Service) attacks.

7.4.2. Integrity of Configuration Information

Two types of integrity must be provided. The first one may be done at the network layer, e.g., by using the IPsec protocol suite. The second type should protect configuration data at the application layer (e.g., the entire file configuration is integrity protected).

7.4.3. Confidentiality of Exchanged Data

The participating device should provide security functions that provide confidentiality. Encryption algorithms must be standard and manually or automatically activated.

7.4.4. Key Management

The configuration system must provide a scalable key management scheme. The number of keys to be managed must be at most linearly proportional to the number of the devices.

7.4.5. Connection Log

The participating device must log all configuration connections. At least the following information must be provided:

- Identity of the device which provided the configuration information,
- Date of the connection,
- Identity of the user who has initiated the configuration process,
- Description of the configuration information that has been forwarded.

7.4.6. Profiles

The configuration system must allow the definition and the activation of several privilege levels. Each level could be associated to a set of administrative functions. Each configuration administrator could be assigned a specific access level to perform a (possibly limited) set of configuration tasks.

8. IANA Considerations

This document does not require any action from IANA.

9. Security Considerations

This document reflects a set of requirements as far as the design and the enforcement of configuration policies are concerned for (automated) service subscription, delivery and maintenance. The

document addresses some security concerns that have been depicted in Section 7.4, and that should be taken into account when considering the specification of a protocol that will convey configuration information, as well as configuration information itself.

10. Acknowledgements

Many thanks to M. Achemlal and Y. Adam who contributed to a first version of this text.

Thanks for W. George for the comments.

11. Informative References

[I-D.boucadair-connectivity-provisioning-protocol] Boucadair, M., Jacquenet, C., Zhang, D., and P. Georgatsos, "Connectivity Provisioning Negotiation Protocol (CPNP)", draft-boucadair-connectivity-provisioning-protocol-08 (work in progress), September 2014.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[RFC2748] Durham, D., Boyle, J., Cohen, R., Herzog, S., Rajan, R., and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.

[RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 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.

[RFC3159] McCloghrie, K., Fine, M., Seligson, J., Chan, K., Hahn, S., Sahita, R., Smith, A., and F. Reichmeyer, "Structure of Policy Provisioning Information (SPPI)", RFC 3159, August 2001.

[RFC3246] Davie, B., Charny, A., Bennet, J., Benson, K., Le Boudec, J., Courtney, W., Davari, S., Firoiu, V., and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, January 2003.

[RFC6022] Scott, M. and M. Bjorklund, "YANG Module for NETCONF Monitoring", RFC 6022, October 2010.

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

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, March 2014.

[RFC7297] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", RFC 7297, July 2014.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Christian Jacquenet
France Telecom
Rennes 35000
France

Email: christian.jacquenet@orange.com

Luis M. Contreras
Telefonica I+D
Ronda de la Comunicacion, s/n
Madrid 28050
Spain

Email: lmcm@tid.es
URI: http://people.tid.es/LuisM.Contreras/