

État de l'art du protocole NETCONF couplé à YANG pour la gestion et l'automatisation d'un parc informatique

Sow Malick p2312542

Encadrant : Thomas Begin

11/02/2026

Résumé

Ce projet porte sur l'étude approfondie du protocole NETCONF et des modèles de données YANG, qui constituent aujourd'hui des technologies centrales de l'automatisation réseau. L'objectif principal est de réaliser un état de l'art des possibilités offertes par ces technologies pour la gestion et l'automatisation d'un parc informatique moderne.

Le travail présenté repose sur une préparation théorique détaillée, une comparaison avec d'autres approches de gestion réseau et l'analyse de scénarios concrets de communication NETCONF. Des références issues de standards, d'ouvrages spécialisés et de documentations constructeurs sont mobilisées afin de situer ce travail dans un cadre académique et industriel reconnu.

1 Introduction

La configuration et la gestion des équipements réseaux reposent historiquement sur des interfaces en ligne de commande (CLI). Cette approche est largement répandue, mais elle devient difficile à maintenir lorsque la taille et la complexité des infrastructures augmentent, en raison du manque de structuration et de validation globale.

Dans un contexte de cloud, de virtualisation et d'automatisation, les réseaux doivent être configurables de manière fiable, reproductible et programmable. NETCONF et YANG s'inscrivent dans cette évolution en proposant une gestion basée sur des données structurées, telle que présentée dans les travaux de référence sur la programmabilité réseau [1].

2 Contexte et état de l'art

2.1 Le protocole NETCONF

NETCONF est un protocole standardisé par l'IETF pour la gestion distante des configurations réseau. Il repose sur une architecture client-serveur et utilise généralement SSH comme transport sécurisé, garantissant la confidentialité et l'intégrité des échanges [2].

Contrairement au CLI, NETCONF introduit des opérations explicites de lecture et d'écriture, ainsi que la notion de datastores distincts¹. Cette approche permet une gestion plus fiable et transactionnelle² des configurations, adaptée aux environnements automatisés et distribués.

2.2 Les modèles de données YANG

YANG est un langage de modélisation permettant de décrire précisément la structure des données manipulées par NETCONF. Les modèles définissent des types, des contraintes et une hiérarchie formelle des paramètres configurables [3].

Cette modélisation permet une validation automatique des configurations avant leur application sur les équipements. Elle réduit significativement les risques d'erreurs humaines et facilite la cohérence globale des configurations dans un parc informatique hétérogène.

2.3 Comparaison avec les autres approches

L'automatisation par scripts CLI est simple à mettre en œuvre, mais elle reste fragile et fortement dépendante des syntaxes constructeur. De plus, elle ne garantit ni l'idempotence ni la validation globale des configurations.

SNMP est historiquement orienté supervision et ne répond pas aux besoins de gestion transactionnelle de configurations complexes. RESTCONF, bien qu'appuyé sur YANG, s'intègre facilement dans des environnements web, mais offre généralement moins de mécanismes avancés que NETCONF pour la gestion complète des configurations [1].

3 Illustration du fonctionnement de NETCONF

Cette section présente des exemples concrets de communication NETCONF afin d'illustrer le fonctionnement réel du protocole. Les échanges reposent sur des messages RPC³ structurés au format XML.

1. Espace logique de stockage de la configuration. Le datastore *running* correspond à la configuration actuellement active sur l'équipement, tandis que le datastore *candidate* permet de préparer une configuration avant sa validation et son application définitive.

2. Approche garantissant qu'un ensemble de modifications de configuration est appliqué de manière atomique, c'est-à-dire entièrement ou pas du tout, afin d'éviter des états incohérents sur l'équipement réseau.

3. Mécanisme permettant à un client d'invoquer à distance des opérations exposées par un serveur, ici sous la forme de requêtes NETCONF structurées en XML.

3.1 Lecture de configuration avec `get-config`

La lecture de configuration constitue une étape essentielle dans tout processus d'automatisation réseau. Elle permet de récupérer l'état courant d'un équipement avant toute modification ou vérification de conformité.

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source><running/></source>
  </get-config>
</rpc>
```

Cette requête interroge un datastore structuré et renvoie des données conformes aux modèles YANG. Contrairement au CLI, les informations retournées sont directement exploitables par un programme sans traitement complexe.

3.2 Analyse d'une réponse NETCONF

La réponse associée à la requête précédente est encapsulée dans un message `rpc-reply`. Les données retournées sont organisées selon une hiérarchie formelle définie par les modèles YANG supportés par l'équipement.

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <interfaces>
      <interface>
        <name>GigabitEthernet0/0</name>
        <enabled>true</enabled>
      </interface>
    </interfaces>
  </data>
</rpc-reply>
```

Cette structuration hiérarchique facilite l'analyse automatique et la vérification de la cohérence de la configuration, ce qui constitue un avantage majeur pour les outils de gestion de parc et d'orchestration.

3.3 Gestion des erreurs avec `rpc-error`

Un aspect fondamental de NETCONF est la gestion structurée des erreurs lors des échanges. Lorsqu'une requête est invalide ou non conforme au modèle YANG, l'équipement renvoie un message `rpc-error` normalisé [2].

```

<rpc-reply message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rpc-error>
    <error-type>protocol</error-type>
    <error-tag>unknown-element</error-tag>
    <error-severity>error</error-severity>
  </rpc-error>
</rpc-reply>

```

Cette structure permet à un programme de détecter précisément la cause du problème et de réagir de manière fiable, ce qui est essentiel dans des scénarios automatisés.

4 Topologie cible et environnement client

Afin de préparer la phase de démonstration, une topologie réseau minimale a été définie. Elle est volontairement simple afin de se concentrer sur les mécanismes de gestion et d'automatisation offerts par NETCONF/YANG plutôt que sur des problématiques de routage avancées.

La topologie envisagée repose sur trois éléments : un routeur, un commutateur et une machine cliente. Le poste client établit une session NETCONF sécurisée avec le routeur via SSH.

4.1 Schéma de la topologie

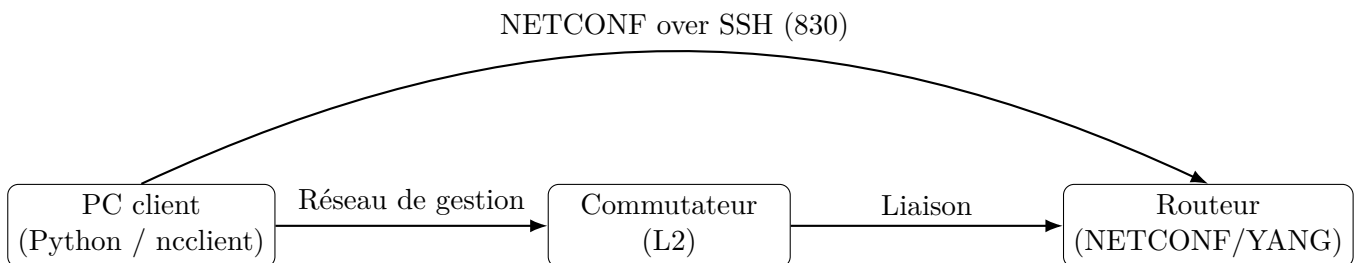


FIGURE 1 – Topologie minimale de démonstration pour l'usage de NETCONF/YANG.

4.2 Outils côté client

La machine cliente héberge les outils nécessaires à la communication NETCONF et à l'automatisation. Un environnement Python est utilisé afin de faciliter l'écriture de scripts et le traitement des réponses XML.

La bibliothèque `ncclient` est couramment utilisée pour implémenter des clients NETCONF en Python. Elle permet de mettre en œuvre rapidement des scénarios de lecture, de modification et de validation de configurations.

5 Analyse et discussion

Les éléments étudiés montrent que NETCONF et YANG offrent une approche plus structurée et plus fiable que le CLI pour l'automatisation réseau. La validation par modèles et la gestion explicite des erreurs constituent des avantages déterminants pour la gestion d'un parc informatique.

Les principales difficultés rencontrées concernent l'environnement logiciel et les mécanismes de sécurité associés. Ces limites soulignent l'importance d'une préparation rigoureuse avant le déploiement d'une automatisation à grande échelle.

6 Conclusion

Ce travail avait pour objectif d'initier un état de l'art autour du protocole NETCONF et des modèles de données YANG dans le contexte de l'automatisation des réseaux. À ce stade, l'étude menée a permis de poser les bases théoriques nécessaires et de clarifier les principes fondamentaux de ces technologies.

Les analyses réalisées et les exemples présentés constituent un socle pour la poursuite de ce travail. La suite du projet consistera à approfondir cet état de l'art et à confronter les concepts étudiés à des expérimentations pratiques, notamment à travers une démonstration réalisée avec l'encadrant du projet et l'ingénieur réseau Romain Grange (responsable de la plateforme réseaux).

7 Références

Références

- [1] M. Clemm, L. Ciavaglia, L. Granville et al., *Network Programmability and Automation*, O'Reilly Media, 2018.
- [2] IETF, *RFC 6241 – Network Configuration Protocol (NETCONF)*, 2011. <https://www.rfc-editor.org/rfc/rfc6241>
- [3] IETF, *RFC 7950 – The YANG 1.1 Data Modeling Language*, 2016. <https://www.rfc-editor.org/rfc/rfc7950>
- [4] Cisco Systems, *NETCONF and RESTCONF Programmability Guide for IOS XE*. <https://www.cisco.com>
- [5] ncclient Documentation, *Python library for NETCONF clients*. <https://ncclient.readthedocs.io>