

L'évolution du SDN avec P4

Oral probatoire en vue d'obtenir l'UE ENG221 « Information et communication pour ingénieur »

Spécialité : INFORMATIQUE — Parcours Architecture et Ingénierie des Systèmes et des Logiciels (A.I.S.L.)

Présentation du 16 juin 2020 à 18 h sur Teams

Sujet n° 37 conduit par le Jury :

PRÉSIDENT : Kamel Barkaoui

MEMBRES : Faten Atigui, Éric Soutil

par Virginie Vélitchkoff

le **cnam**
EICnam

Définition du sujet

L'évolution du SDN avec P4

Le SDN (Software-Defined Networking) est un nouveau paradigme permettant une programmation de la pile réseau et favorisant sa virtualisation. Récemment, ce paradigme a été revu avec le « Next-Gen-SDN » qui essentiellement vise à ne pas mettre des limites aux opérations qu'un commutateur peut effectuer dans le traitement du trafic. Une brique essentielle de cette évolution est le langage P4 qui permet une programmation des commutateurs tout en donnant des garanties de rapidité dans l'exécution du programme compilé et installé dans les commutateurs SDN. Il s'agit de présenter cette évolution du SDN et de toutes les nouvelles briques logicielles impliquées dans l'opération d'un réseau avec P4.

Mots-clés : SDN, P4, NFV

Bibliographie d'entrée :

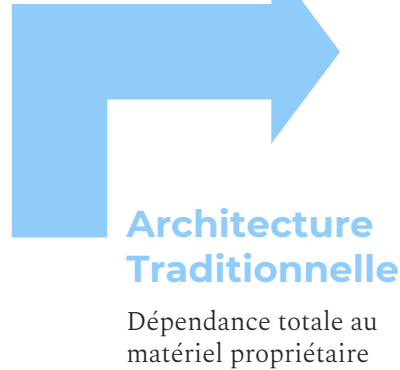
<https://www.opennetworking.org/p4/>

Bosshart, Pat, et al. "P4 : Programming protocol-independent packet processors." ACM SIGCOMM Computer Communication Review 44.3 (2014): 87-95.

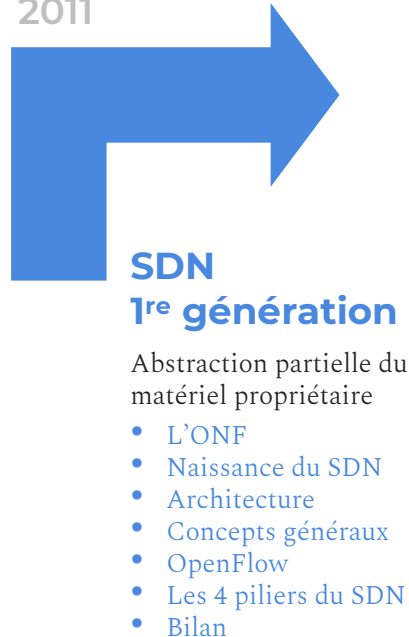
Sommaire

Vue d'ensemble de l'évolution du SDN

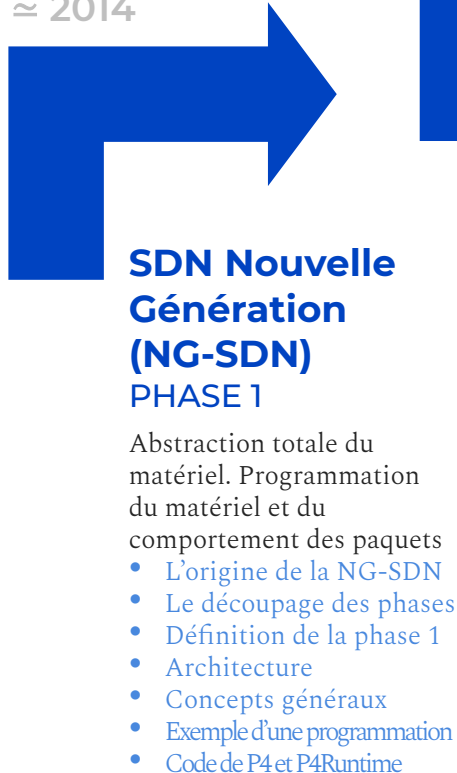
À l'origine



2011



≈ 2014



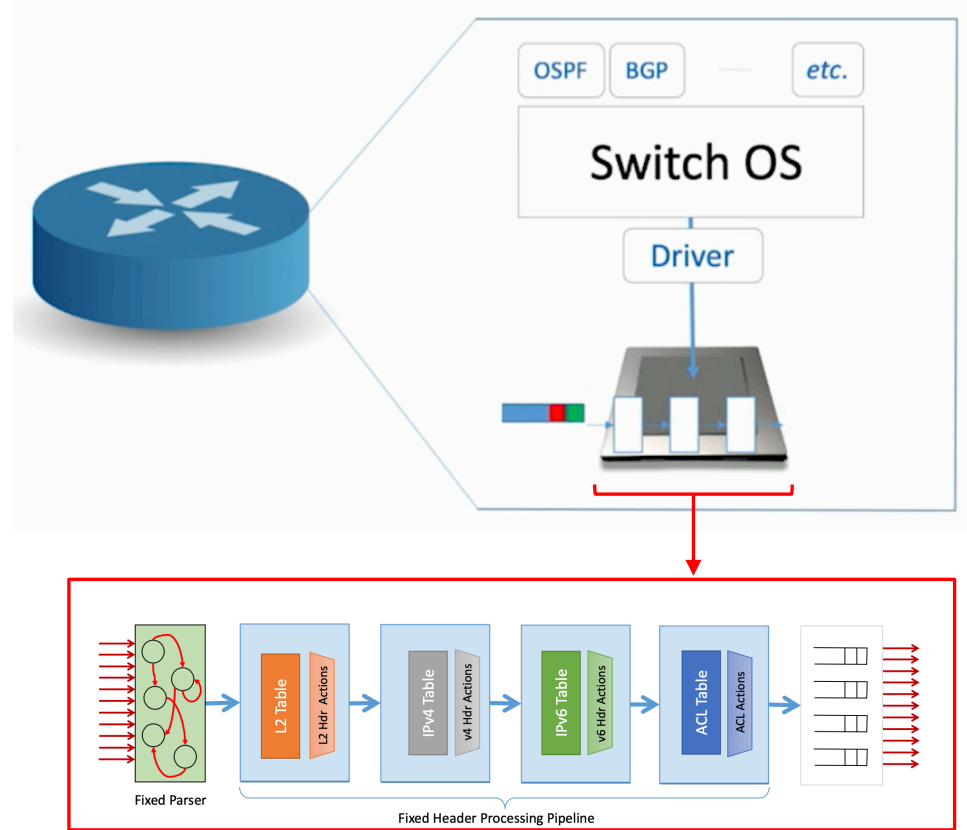
2030



Architecture Traditionnelle

Un commutateur est un matériel propriétaire à l'architecture verticale de bas en haut qui intègre plan de contrôle et plan de commutation en un seul et même équipement.

Cette rigidité ralentissait l'innovation : l'implantation d'un nouveau protocole pouvait prendre jusqu'à 10 ans.



L'Open Networking Foundation (ONF)

Mission

Créé en 2011, l'ONF est un consortium à but non lucratif dirigé par des opérateurs renommés pour la transformation de l'infrastructure réseau répondant aux modèles commerciaux des opérateurs.

L'ONF est composé d'un laboratoire de recherche dédié et d'une équipe orientée sur les décisions stratégiques.

Visions

- Accélérer et généraliser la participation à l'innovation pour répondre aux besoins toujours plus ambitieux et plus grands des réseaux (data center, réseaux télécoms, réseaux virtuels, sans fil...)
- Augmenter les revenus en réduisant les coûts à tous les niveaux
- Former, promouvoir, inciter les opérateurs à déployer les solutions innovantes de l'ONF

Valeurs

L'ONF est une communauté de communautés collaboratives et open source.

Standards
approuvés
et publiés



Définition

Software- Defined Networking SDN



Un réseau dans lequel le plan de contrôle est physiquement séparé du plan de commutation et un seul plan de contrôle contrôle plusieurs équipements. »¹

—Nick McKeown's 2013 presentation entitled Software Defined Networking.

¹ "A network in which the control plane is physically separate from the forwarding plane, and a single control plane controls several forwarding devices."

1^{re} génération SDN



OBJECTIFS

Accélérer l'innovation
Réduire les coûts

Favoriser l'adéquation de l'innovation des réseaux et celle du matériel, réduisant ainsi les coûts et augmentant les profits et le temps de mise sur le marché.
(ex. : Ajouter un nouveau protocole à un équipement prenait jusqu'à 10 ans. À ce rythme, le protocole pouvait être déjà obsolète)



IDÉE

Prendre exemple sur
un ordinateur!

Peu importe sa marque et ses caractéristiques, un système d'exploitation tourne dessus grâce à une couche de virtualisation qui permet de s'abstraire de l'infrastructure et d'offrir des fonctionnalités diverses

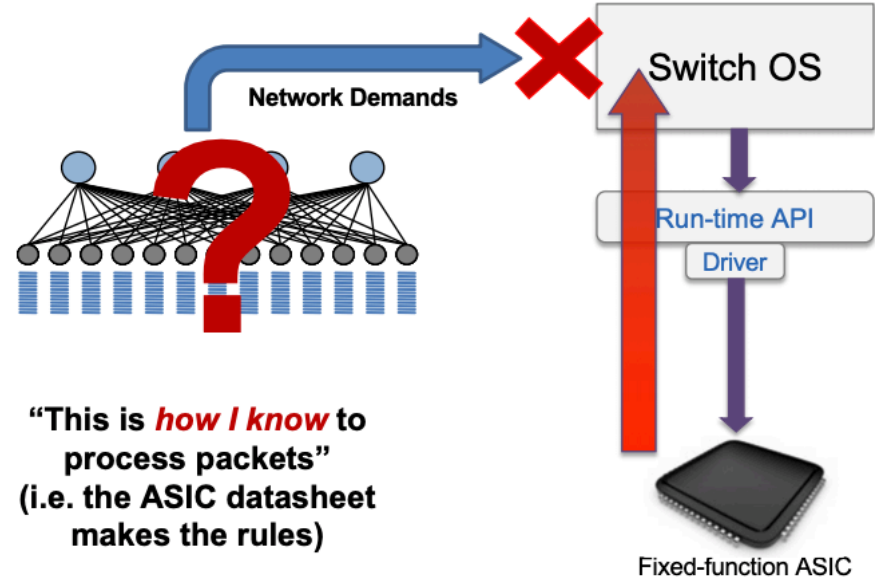
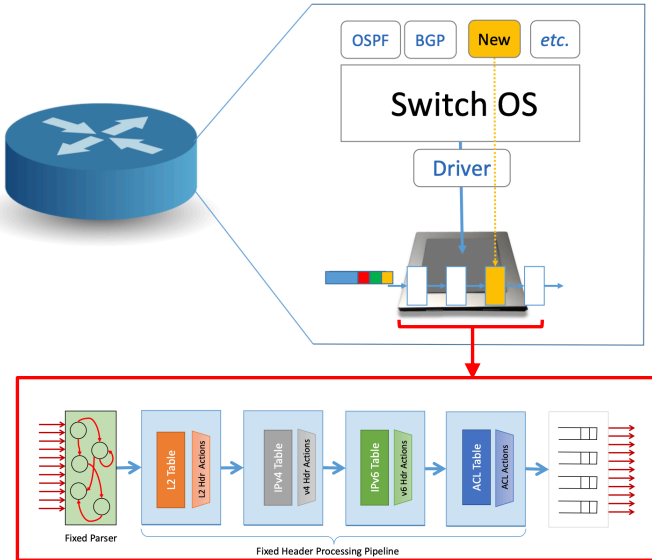


RÉALISATION

Séparer l'intelligence
logicielle du matériel

L'API OpenFlow permet l'abstraction du matériel en proposant une table fixe d'entêtes compatibles avec tous les équipements (en se basant sur ces derniers). Ainsi, le SDN découple le plan de commutation, du plan de contrôle permettant une forte isolation des couches

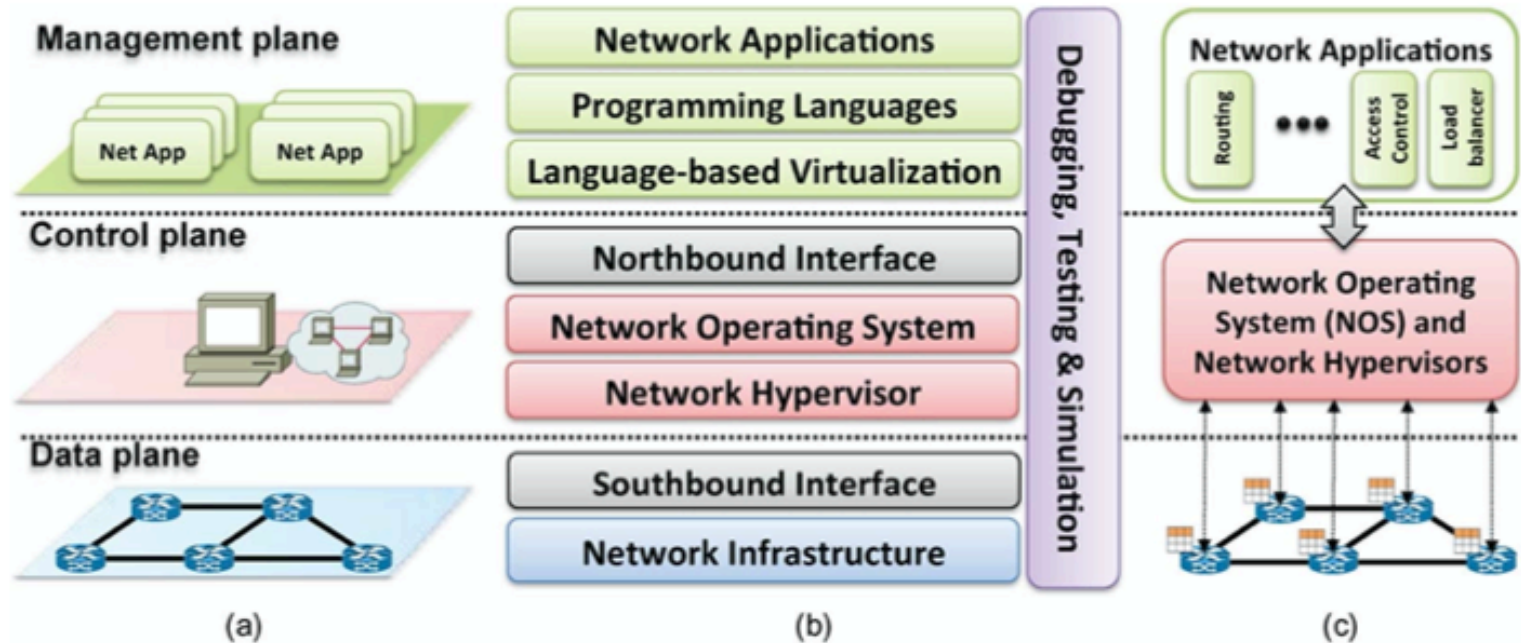
Architectures Software-Defined Network



Sources :

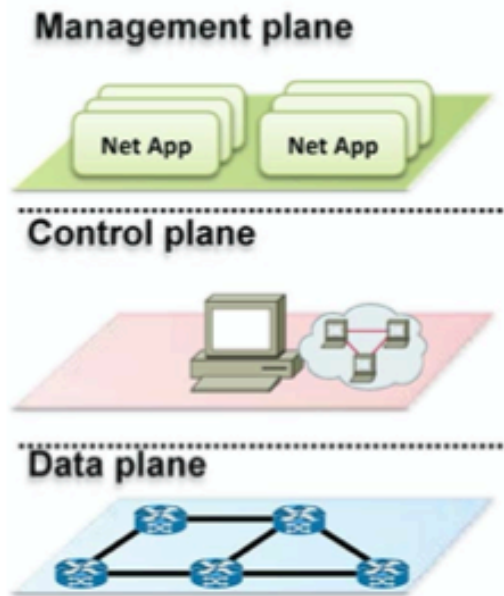
ONF Connect 2019, How We Might Get Humans Out of the Way - Keynote by Nick McKeown
P4.org, « P4 Language Overview », <https://cs344-stanford.github.io/lectures/Lecture-2-P4-tutorial.pdf>

Architectures Software-Defined Network



Source : Kreutz et al. (2015), Software-Defined Networking : A Comprehensive Survey, Software-Defined Networks in (a) planes, (b) layers, and (c) system design architecture.

Concepts généraux : 1^{re} génération SDN



Plan de gestion (Management Plane)

Détermine une politique de plan de contrôle centralisé qui sera renforcé par les plans de contrôle. Son rôle est de permettre de déployer de nouvelles fonctionnalités réseau à travers des applications en fonction des ressources.

Plan de contrôle (Control Plane)

Définit comment gérer le trafic réseau et ses équipements en fonction de la politique du plan de gestion. Il est composé de plusieurs plans de contrôle distribués qui gèrent le flux de données.

Ses missions :

- Définir une politique de traitement des paquets de données
- Peupler les tables de routage des plans de commutations
- Configurer ou éteindre les plans de commutation.

Plan de commutation (Data Plane)

Est dans le matériel. Son rôle est d'exécuter les décisions du plan de contrôle tel que :

- Rechercher les données
- Traiter, aiguiller et mettre en action la politique de gestion de plan de contrôle
- Collecter des statistiques

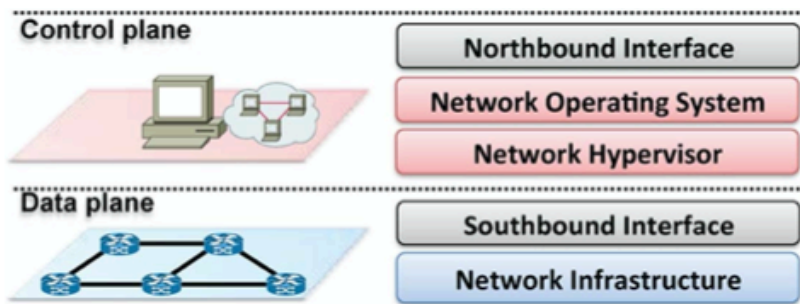
L'API OpenFlow pour l'abstraction du matériel

Son principe

Il constitue l'API Southbound qui fait le lien entre la couche 2 (système d'exploitation réseau) et la couche 3 (d'infrastructure).

Ce protocole est un élément fondamental pour la construction de solutions SDN puisqu'il permet de programmer un plan de contrôle pour n'importe quel équipement et logiciels réseau.

Sans lui, pas d'abstraction de l'équipement.

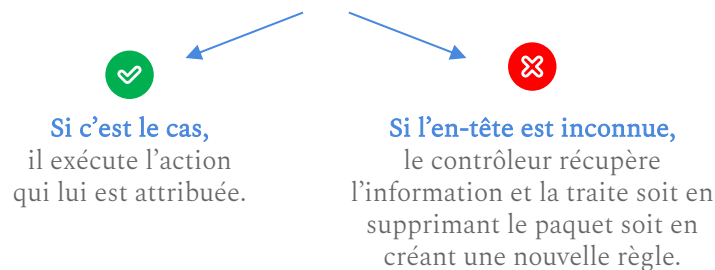


Son architecture

OpenFlow est intégré dans le commutateur où ils possèdent plusieurs tables de flux qui contiennent chacune des règles (composées d'en-têtes, d'action et de compteurs statistiques). Ces règles définissent des sous-ensembles de données à traiter auxquelles sont attribuées des actions.

Son fonctionnement

Un paquet entre dans le commutateur qui vérifie que son en-tête est contenu dans l'une des règles de ces tables.



1^{re} génération SDN – Les 4 piliers

Le plan de contrôle et de commutation découplé

Le plan de commutation intégré à l'appareil devient un simple élément de transmission de données alors que le plan de contrôle en est retiré.

Les décisions de transmissions sont basées sur le flux et non sur la destination

Un flux est défini par un ensemble de paquets avec un critère de correspondance et des instructions ou actions qui s'y rapportent.

La logique de contrôle est déplacée vers une entité externe : le contrôleur SDN ou NOS (Network Operating System)

Le NOS est une plateforme logicielle qui fonctionne sur la technologie des serveurs de base et fournit les ressources et abstractions essentielles pour faciliter la programmation des périphériques de transfert sur la base d'une vue de réseau abstraite et centralisée logiquement. Son objectif est donc similaire à celui d'un système d'exploitation traditionnel.

Le réseau est programmable grâce à des applications logicielles exécutées au-dessus du NOS

Il s'agit d'une caractéristique fondamentale de SDN, considérée comme sa principale proposition de valeur.

Bilan de la 1^{re} génération SDN



Administration du réseau centralisée

Le plan de gestion couplé au plan de contrôle permet une gestion globale, intelligente, sécurisée, optimisée, voire automatisée du réseau.



Plan de contrôle programmable

Le plan de contrôle gère et contrôle plusieurs appareils par des API.

Il permet un traitement par flux de données et non par destination.

Il a une capacité d'adaptation et de personnalisation aux nouveaux protocoles.



Découplage

Permet d'abstraire le matériel et le plan de commutation sans impacter sur la politique du plan de contrôle.

Ainsi chaque couche isolée de l'architecture a son propre rôle qui n'impacte pas les autres en cas de changement.



Domaines d'application

Le SDN répond à des problématiques multiservices, multiprotocoles et multitechnologies tels que :

- **Big Data** : possibilité de reconfiguration à volonté
- **Machine Learning** : prévision du trafic, sécurité...
- **5G, IoT et Smart Cities** : scalabilité

L'origine de la Prochaine Génération SDN (NG-SDN)



Problématique

Le SDN et OpenFlow a été largement adopté, des fournisseurs comme Google l'ont implanté et mis en production. D'autres problématiques ont surgi :

- **L'impossibilité de continuer à faire évoluer les tables de flux d'OpenFlow** motivant la naissance de P4 et P4Runtime
- **Repenser l'architecture** verticale du haut vers le bas, c'est-à-dire depuis le système d'exploitation vers le matériel afin de s'abstraire complètement du matériel et non partiellement comme avec OpenFlow
- **Contrôler le traitement des paquets.**



Solutions

Phase 1

Un changement d'architecture avec la révolution P4 et P4Runtime pour l'abstraction complète de l'équipement et la propagation de la programmation dans toutes les strates du domaine du réseau. Cette phase a commencé et est toujours en cours.

Phase 2

L'indépendance des réseaux programmables avec la télémétrie, le CI/CD, la conteneurisation et l'orchestration et la vérification à toutes les étapes... Initiée depuis 2020, elle devrait aboutir en 2030...

Définition

Nouvelle génération SDN (NG-SDN)

Cette nouvelle génération a pour but de changer tout l'écosystème du réseau en le rendant programmable de bout en bout afin de pouvoir l'automatiser.



Une plateforme open source SDN de nouvelle génération (NG-SDN) intégrant plusieurs technologies pour fournir une solution de réseau indépendante du matériel, programmable, vérifiable, sans contact humain pour sa gestion et sa configuration. Le tout reposant sur des logiciels open source. ”

— ONF

Origine et évolution des phases 1 et 2 du **NG-SDN**



OBJECTIFS



IDÉES



OUTILS

Phase 1

- S'abstraire complètement du matériel
- Contrôler le traitement des paquets indépendamment des équipements

- Inverser l'architecture afin que l'OS puisse s'adapter à n'importe quel équipement
- Fournir une implémentation dynamique de protocoles et de pipelines indépendante du matériel
- Créer un langage de programmation pour pouvoir définir le comportement des paquets, leur processus

- P4
- P4Runtime

Phase 2

- Rendre le fonctionnement du réseau de plus en plus indépendant du contrôle humain

- Créer les outils qui permettront de monitorer le réseau tout en faisant évoluer l'architecture :
 - nouvelles API
 - systèmes d'exploitation
 - télémétrie et système d'observation
 - Systèmes de Vérifications à tous les niveaux
 - CI/CD
 - Conteneurisation et orchestration...

- ONOS
- STRATUM
- Moteur de vérification
- Infrastructure de développement d'opérationnalisation (CI/CD)

NG-SDN

Phase 1, aujourd'hui

Cette phase est l'amorce d'un changement de tout l'écosystème du réseau qui devient programmable : **les programmeurs prennent le contrôle** sur le traitement des paquets dans le plan de commutation.



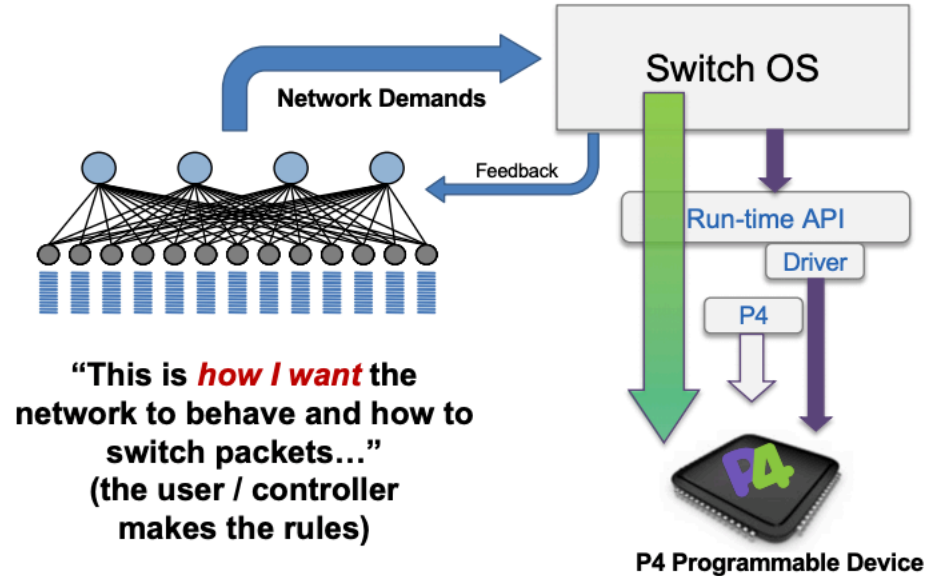
L'innovation doit venir des programmeurs et non plus des équipementiers.”

—Nick McKeown's Keynote 2019

Architecture NG-SDN Phase 1

L'architecture s'inverse : les réseaux sont définis de haut en bas par les opérateurs et/ou les développeurs d'applications et non plus de bas en haut d'après les ASIC.

La révolution P4 et P4Runtime permet cette abstraction complète de l'équipement et la propagation de la programmation dans toutes les strates du domaine du réseau. Cette phase a commencé et toujours en cours.



Copyright © 2019 – P4.org

Source : P4.org, « P4 Language Overview ». <https://cs344-stanford.github.io/lectures/Lecture-2-P4-tutorial.pdf>

Concepts généraux de la NG-SDN Phase 1

P4

Bosshart et al. (2014) crée **le langage open source dédié de haut niveau P4** (Programming Protocol-Independent Packet Processors) destiné à faire correspondre les paquets avec n'importe quels champs d'en-têtes.

Principales propriétés de ce langage :

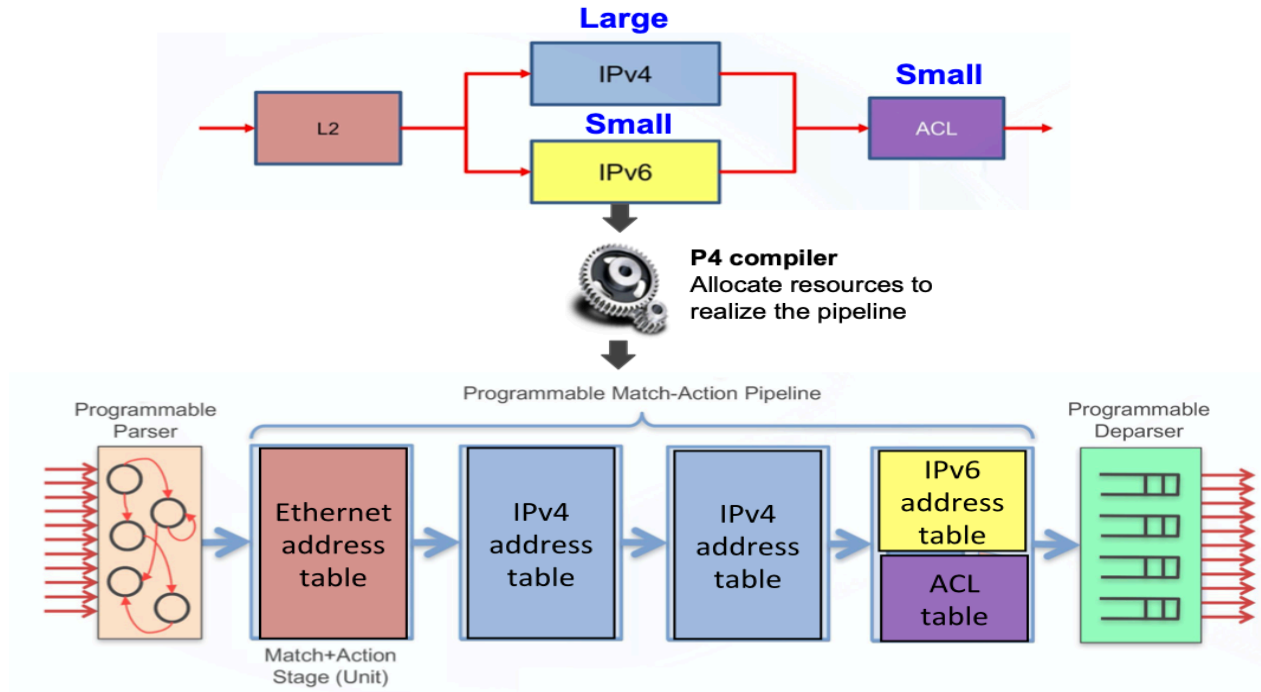
- **Définir formellement le pipeline** du plan de commutation tel que : les en-têtes de protocole, les tables de correspondances, les actions, les statistiques, etc.
- **Configurer la vitesse du pipeline** selon l'équipement qui lui est destiné
- **Assurer la rétroactivité** avec les commutateurs à fonction fixe

P4Runtime

Initiées par L'ONF et Google en 2016, les premières spécifications sont sorties en 2019. **L'API P4Runtime est une spécification de plan de contrôle pour contrôler les éléments du plan de commutation d'un périphérique défini ou décrit par un programme P4.**

- **P4Runtime est le seul outil qui propose l'indépendance des protocoles** : P4Runtime peut contrôler le plan de commutation de n'importe quel protocole standard ou personnalisé
- **L'API est indépendante des commutateurs et de ses fournisseurs et indépendante des pipelines** de plan de commutation qui ont été spécifiquement formalisés et programmés en P4
- **L'API se met automatiquement à jour dès que de nouvelles fonctionnalités sont introduites dans un programme P4** et est exposée à un schéma de contrôle externe. La programmation est dynamique et cela nous donne une API unique qui peut être utilisée pour contrôler plusieurs interrupteurs polyvalents
- **L'API permet de mieux contrôler des contrôleurs distribués**
- **P4Runtime est le cœur de l'interface de Stratum**

Compiler un programme P4 sur un commutateur programmable



Code d'un programme P4 et P4Runtime

P4Info example

20

basic_router.p4

```
...  
action ipv4_forward(bit<48> dstAddr,  
                    bit<9> port) {  
    eth.dstAddr = dstAddr;  
    metadata.egress_spec = port;  
    ipv4.ttl = ipv4.ttl - 1;  
}  
...  
table ipv4_lpm {  
    key = {  
        hdr.ipv4.dstAddr: lpm;  
    }  
    actions = {  
        ipv4_forward;  
        ...  
    }  
    ...  
}
```



P4 compiler

basic_router.p4info

```
actions {  
  id: 16786453  
  name: "ipv4_forward"  
  params {  
    id: 1  
    name: "dstAddr"  
    bitwidth: 48  
    ...  
    id: 2  
    name: "port"  
    bitwidth: 9  
  }  
}  
...  
tables {  
  id: 33581985  
  name: "ipv4_lpm"  
  match_fields {  
    id: 1  
    name: "hdr.ipv4.dstAddr"  
    bitwidth: 32  
    match_type: LPM  
  }  
  action_ref_id: 16786453  
}
```

Copyright © 2019 - Open Networking Foundation

P4Runtime table entry WriteRequest example

21

basic_router.p4

```
action ipv4_forward(bit<48> dstAddr,  
                    bit<9> port) {  
    /* Action implementation */  
}  
table ipv4_lpm {  
    key = {  
        hdr.ipv4.dstAddr: lpm;  
    }  
    actions = {  
        ipv4_forward;  
        ...  
    }  
}
```

Logical view of table entry

hdr.ipv4.dstAddr=10.0.1.1/32
-> ipv4_forward(00:00:00:00:00:10, 7)

Control plane
generates

WriteRequest message (protobuf text format)

```
device_id: 1  
election_id { ... }  
updates {  
  type: INSERT  
  entity {  
    table_entry {  
      table_id: 33581985  
      match {  
        field_id: 1  
        lpm {  
          value: "\n\000\001\001"  
          prefix_len: 32  
        }  
      }  
      action {  
        action_id: 16786453  
        params {  
          param_id: 1  
          value: "\000\000\000\000\000\000\n"  
        }  
        params {  
          param_id: 2  
          value: "\000\007"  
        }  
      }  
    }  
  }  
}
```

Copyright © 2019 - Open Networking Foundation

NG-SDN

Phase 2, le futur

Cette nouvelle phase veut fournir une solution de réseau **toujours plus indépendante du matériel, programmable, vérifiable et sans contact humain**. Des technologies ont déjà été mises en place, d'autres sont à l'état de recherche stratégique.



“Les réseaux seront [espérons-le] **programmés par beaucoup** et **exploités par peu**.”¹

—Nick McKeown's

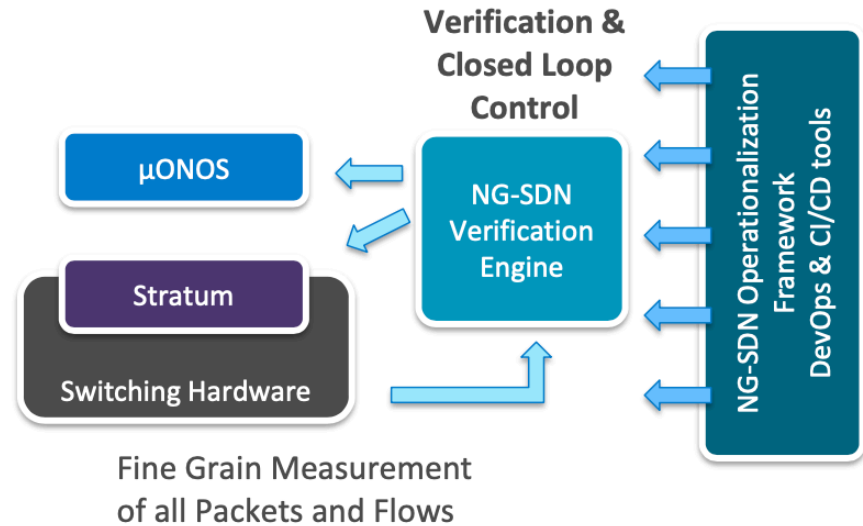
¹ “Networks will [hopefully] be programmed by many, and operated by few.”

Architecture NG-SDN Phase 2

Les composants open source de cette nouvelle architecture comprennent :

- μONOS (une évolution de ONOS)
- STRATUM
- Moteur de vérification
- Infrastructure de développement d'opérationnalisation (CI/CD)

L'ONF a créé des **commutateurs de boîte blanche** dans le but de limiter les coûts et de ne pas dépendre d'un matériel propriétaire onéreux.



Concepts généraux de la NG-SDN Phase 2

STRATUM

Stratum est un système d'exploitation de distribution légère qui intègre le minimum des interfaces : P4Runtime, OpenConfig et des outils de configuration et de télémétrie (gNMI et gNOI). Il est destiné aux **commutateurs de boîte blanche** de l'ONF et supporté par quelques fournisseurs.

Moteur de vérification NG-SDN

Une vérification continue du comportement précis du réseau **offre la garantie du comportement spécifié, monitoré.**

ONOS

Onos une plateforme de contrôle et de configuration cloud native SDN qui gère un réseau de commutateurs Stratum de manière **logiquement centralisée.**

ONOS prend en charge à la fois la configuration et le contrôle en temps réel du réseau : plus de protocoles de contrôle de routage et de commutation.

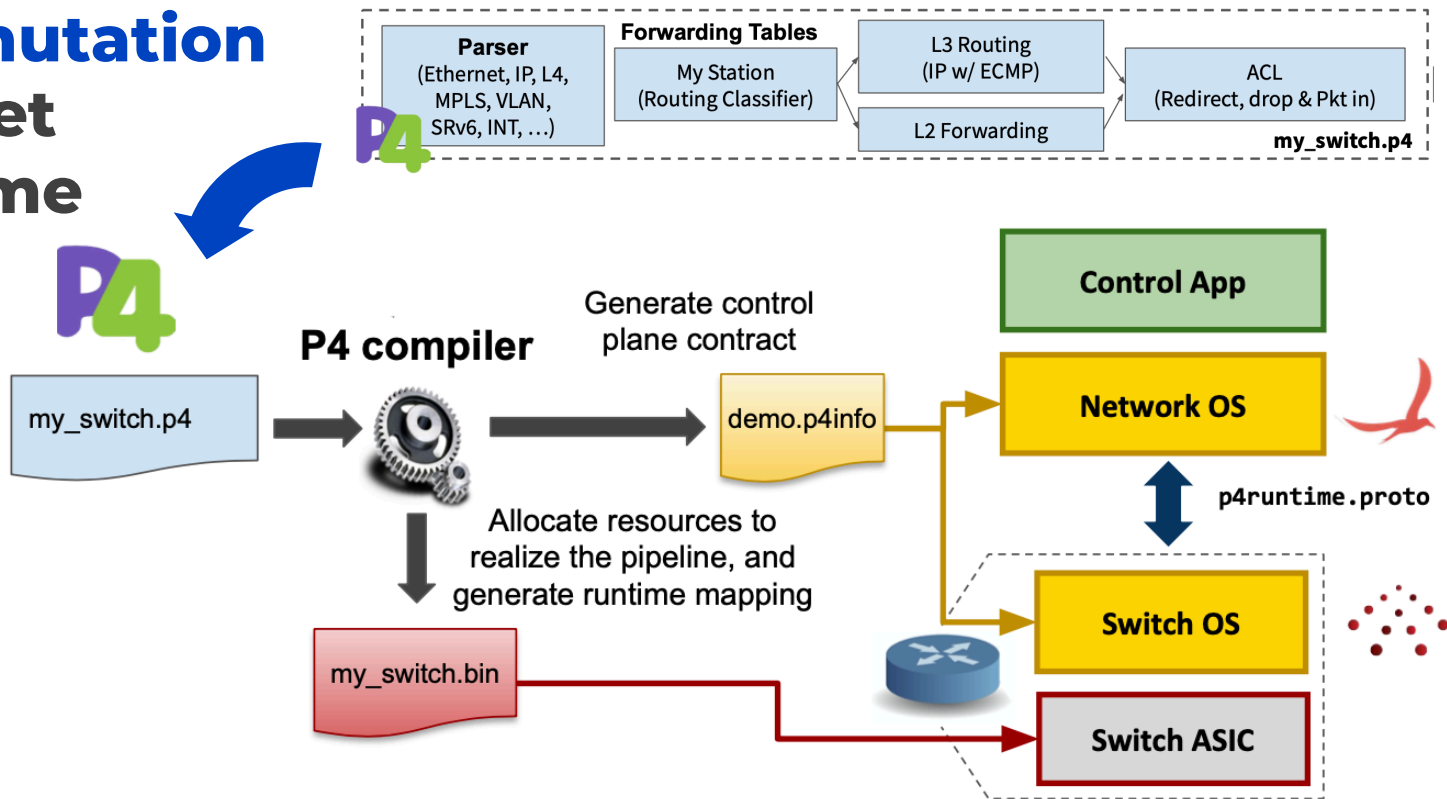
Les utilisateurs finaux peuvent facilement créer de nouvelles applications réseau sans avoir à modifier les systèmes de plan de commutation.

Infrastructure de développement d'opérationnalisation

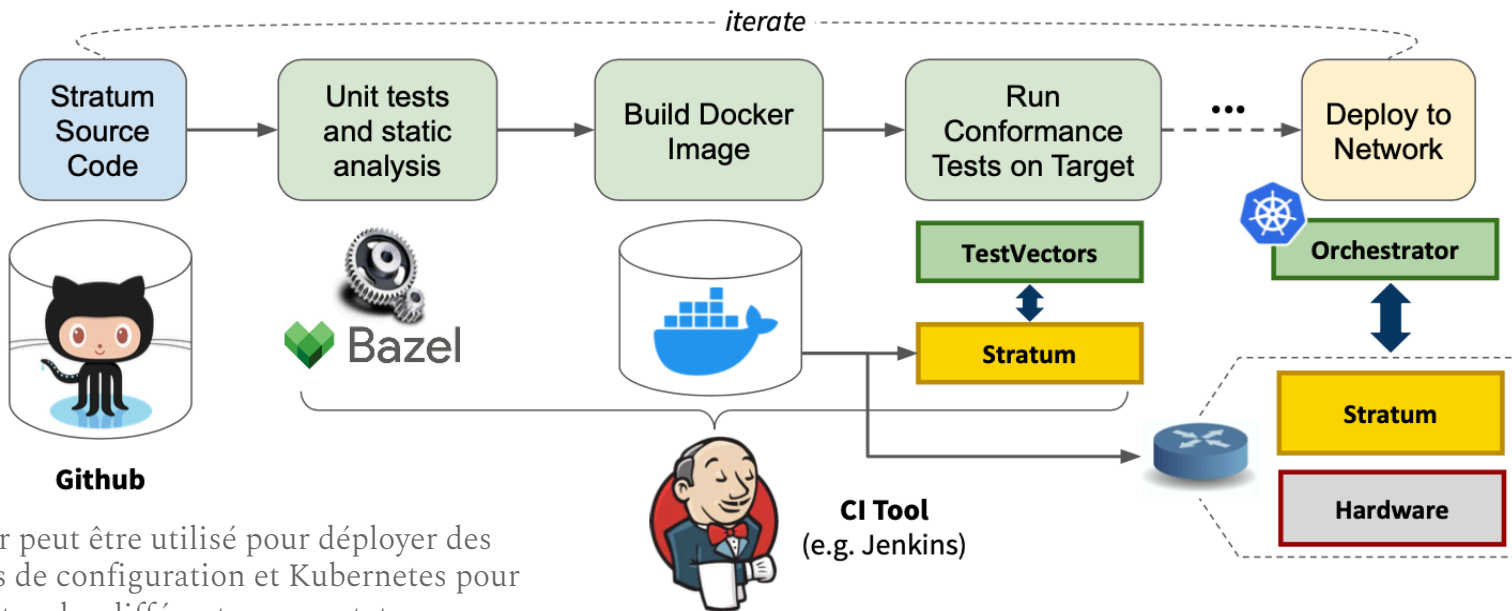
Le réseau est géré selon la méthode **Continuité d'intégration et de déploiement (CI/CD)**, permettant des mises à jour rapides, des déploiements automatisés, une vérification et des restaurations automatisées.

Le réseau devient une ressource dynamique et agile qui peut être exploitée et mise à jour sans craindre que les changements ne dégradent l'infrastructure de l'opérateur.

Définir un plan de commutation avec P4 et P4Runtime

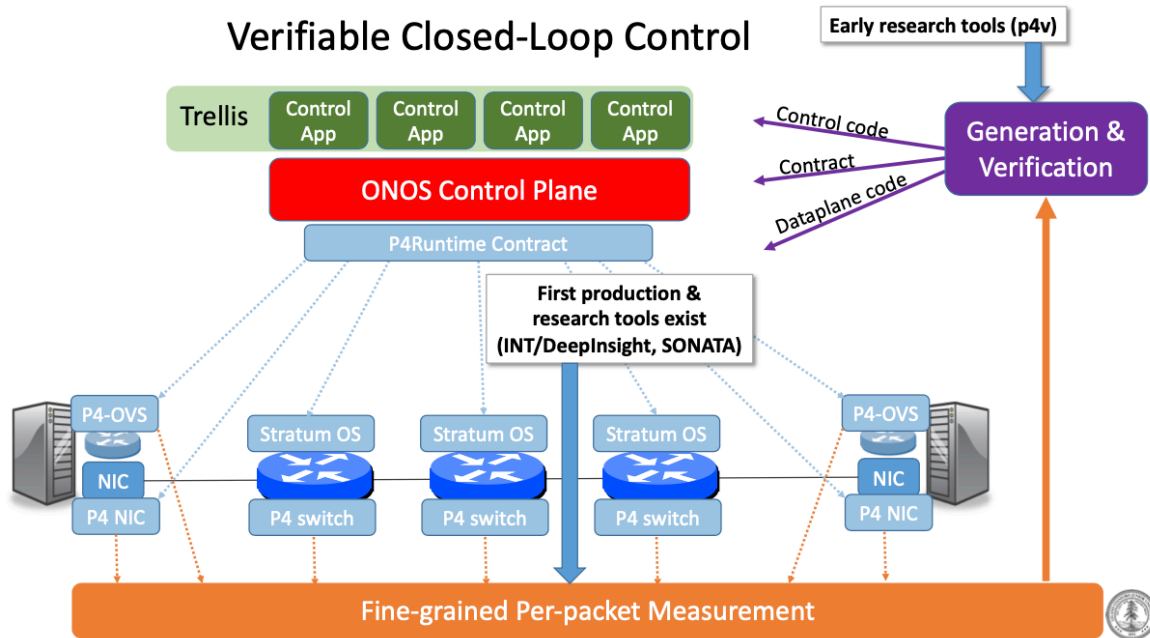


Infrastructure de développement en intégration continue et déploiement continu (CI/CD)

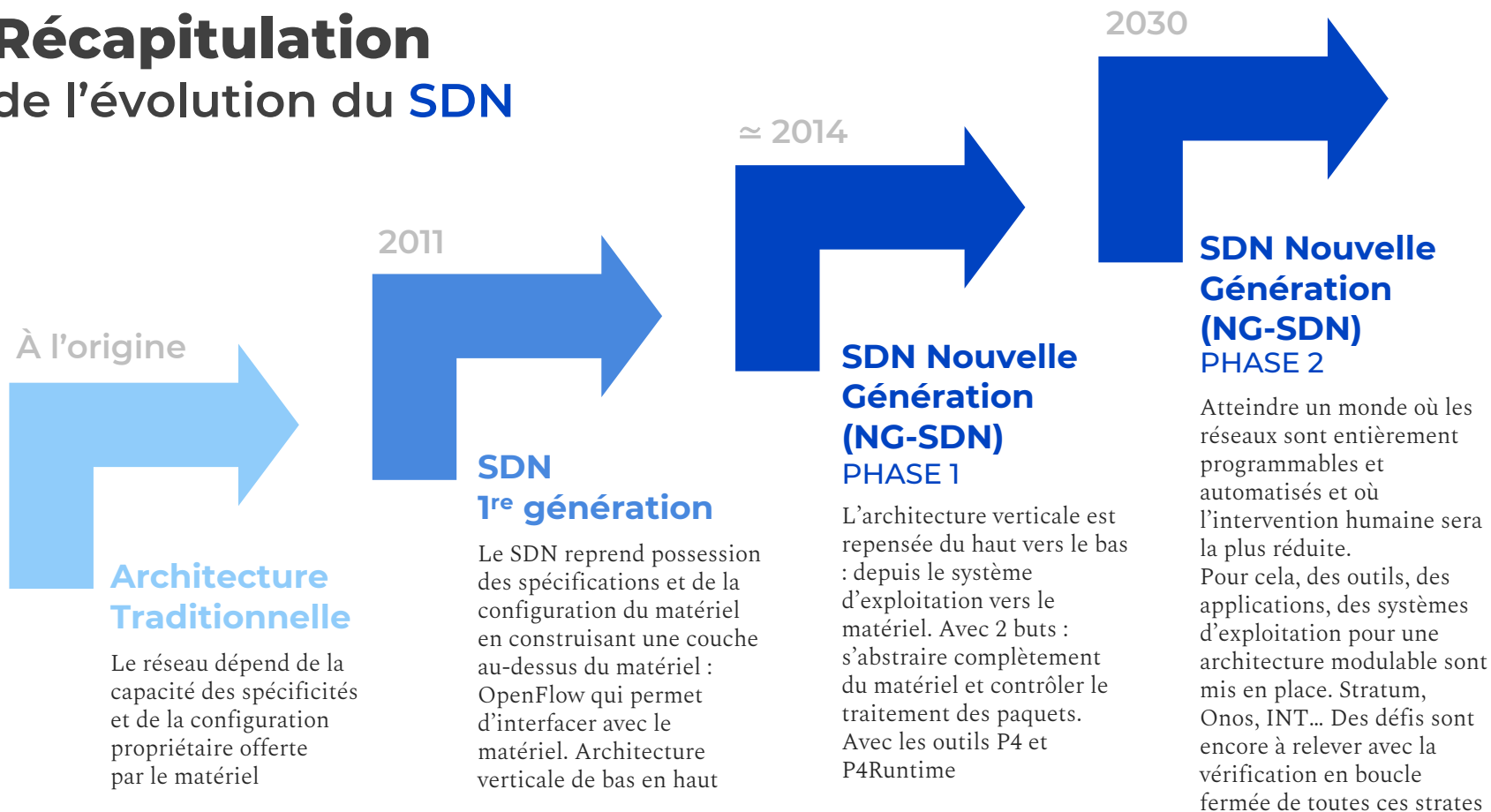


Docker peut être utilisé pour déployer des images de configuration et Kubernetes pour orchestrer les différents commutateurs comme le montre l'exemple ci-dessous.

Phase 2 : Le futur à développer, la vérification et le contrôle en boucle fermé



Récapitulation de l'évolution du SDN



Conclusion

Le SDN, toutes générations confondues, a pour but de séparer le matériel de l'intelligence logicielle et de favoriser l'innovation et la recherche en permettant une réaction plus rapide entre le temps de conception et de réalisation ("time to market").

Même si le développement du SDN est en cours, on peut lui reconnaître des ambitions considérables dans des délais qui peuvent sembler courts. Toutefois, la vision et la volonté dans la communauté des opérateurs et des autres acteurs du réseau sont assez frappantes et elles ont déjà apportées beaucoup de progrès, d'innovation, de succès et surtout l'adoption des solutions.

On ne peut qu'espérer que les objectifs séduisants et pour l'instant rêveurs de Nick McKeown et de L'ONF continuent de se réaliser...

Bibliographie

- [1] « Wikipédia:Accueil principal », *Wikipedia, the free encyclopedia*. nov. 08, 2019, Consulté le: juin 14, 2020. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Wikip%C3%A9dia:Accueil_principal&oldid=164303621.
- [2] « Wikipédia: Main Page », *Wikipedia, the free encyclopedia*. mai 15, 2020, Consulté le: juin 14, 2020. [En ligne]. Disponible sur: https://en.wikipedia.org/w/index.php?title=Main_Page&oldid=956868703.
- [3] P. Bosshart *et al.*, « P4: programming protocol-independent packet processors », *SIGCOMM Comput. Commun. Rev.*, vol. 44, n° 3, p. 87–95, juill. 2014, doi: 10.1145/2656877.2656890.
- [4] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, et S. Uhlig, « Software-Defined Networking: A Comprehensive Survey », *Proceedings of the IEEE*, vol. 103, n° 1, p. 14-76, janv. 2015, doi: 10.1109/JPROC.2014.2371999.
- [5] O. N. Foundation, *How We Might Get Humans Out of the Way - Keynote by Nick McKeown, Professor at Stanford University - ONF Connect 19*. 2019. <https://vimeo.com/359434741> (Consulté le 08/06/2020)
- [6] O. N. Foundation, *Next-Gen SDN Stack Plenary - ONF Connect 19*. 2019. <https://vimeo.com/359680444> (Consulté le 14/06/2020)
- [7] V. Velitchkoff, *My playlist of SDN presentations - YouTube*. <https://www.youtube.com/playlist?list=PLB3XWQvxupiqnOteeHIhn56acCShowKn0> (Consulté le 07/06/2020)
- [8] ONS Americas 2019: *P4 and P4Runtime Technical Introduction and Use Cases for the Next-Gen SDN Stack*. . <https://www.youtube.com/watch?v=q-vwFS8dJBM&list=PLB3XWQvxupiqnOteeHIhn56acCShowKn0&index=2&t=194s> (Consulté le 14/06/2020)
- [9] P4.org, « P4 Language Overview ». <https://cs344-standford.github.io/lectures/Lecture-2-P4-tutorial.pdf> (consulté le juin 14, 2020).
- [10] « Chapter 1: Introduction — Software-Defined Networks: A Systems Approach Version 0.3-dev documentation ». <https://sdn.systemsapproach.org/intro.html#sdn-a-definition> (consulté le juin 14, 2020).
- [11] « NG-SDN Definition », *Open Networking Foundation*. <https://www.opennetworking.org/ng-sdn/> (consulté le juin 15, 2020).

Merci aux professeurs du Cnam qui m'ont donnés les clés et les enseignements pour mener à bien ce projet.

Merci à une amie pour m'avoir aidé à maîtriser la suite Microsoft office afin de mener à bien ma présentation.

Pour toutes questions complémentaires,
je reste à votre disposition,

- Virginie Vélitchkoff
- virgcnam@gmail.com
- <https://www.linkedin.com/in/hellov/>

Merci
