

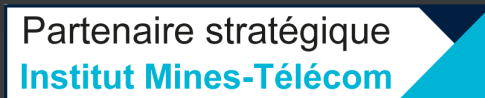
# Réseaux Programmables

## Rapport de projet

Perfectly Optimized Kit for Efficient Monitoring  
(POKEMON)

Brelot Julien

*Git*





# Table des matières

I	Introduction	.....
II	Réalisations	.....
	II.1 Contenu	.....
	II.2 Fonctionnement	.....
III	Conclusion	.....
I	Annexe	.....
	I.1 Déploiement du Code	.....

Ce rapport présente le travail réalisé dans le cadre du mini-projet de Réseaux Programmables.

L'objectif de ce projet est de mettre en place un réseau d'équipements réseaux programmables, avec une solution de supervision et de contrôle des liens.

Dans ce but, le data plane est programmé en P4 avec un fichier source unique et générale pour l'ensemble des équipements.

Le control plane est programmé en Python et permet de contrôler les équipements et de collecter les informations de supervision à l'aide d'un méta-contrôleur.

Les routeurs disposent de 3 types controle plane différent : simple, avec perte et une dernier définissant un routage non optimal.

## II Réalisations

### II.1 Contenu

On retrouve dans ce mini-projet :

- Un fichier P4 pour le data plane des 3 types de routeurs.
- Un méta-contrôleur qui permet de contrôler les routeurs et de collecter les informations de supervision.
- 3 types de contrôle plane définis en python pour les routeurs : *simple\_router*, *simple\_router\_oss* et *simple\_router\_tupid*.
- Un fichier python network.py qui permet de générer et lancer un réseau.

### II.2 Fonctionnement

**Data Plane**

## III Conclusion

Cette étude met en évidence les avantages et inconvénients des architectures NoSQL et relationnelles dans divers scénarios.

MongoDB offre une grande flexibilité et scalabilité, toutefois, la réplication et le sharding peuvent impacter les performances des opérations CRUD. Néanmoins, le sharding, s'il est possible est une très bonne solution.

Avec une meilleure configuration, on pourrait s'attendre à des performances similaires au mode standalone et on a tout de même obtenu des meilleures performances que dans le mode répliqué.

Toutefois MySQL, reste performant pour des cas nécessitant des transactions complexes et ne peut montrer l'étendue de ses performances dans un cas aussi simple ne nécessitant pas de modèle relationnel.

De plus, les performances du cluster MySQL s'est avéré, contre toute attente, plus performant que MySQL Standalone, et aussi performante que MongoDB Standalone.

# I Annexe

## I.1 Déploiement du Code

Le projet peut être cloné depuis le dépôt git suivant :

```
git clone https://github.com/BJCode-git/Projet-TDLE.git -b main &&  
cd Projet-TDLE
```

*Installation des dépendances python :*

```
pip3 install -r requirements.txt
```

*Démarrage du réseau :*

```
python3 network.py
```

*Démarrage du méta-contrôleur et des simple router :*

```
python3 controllers/meta_controller.py
```