

# Tests de Performance des systèmes embarqués de la voiture de demain

Mohamed OUMRI - Marc HAGE CHAINE

09.04.2019

**altran**



# Agenda

- 1** Contexte des tests de performance et architecture des systèmes à tester
- 2** Tests de performance et prédiction des performances par réseau de neurones
- 3** Conclusion et perspectives



# Le réseau de communication à bord des véhicules de demain

## Le projet CoherenSE

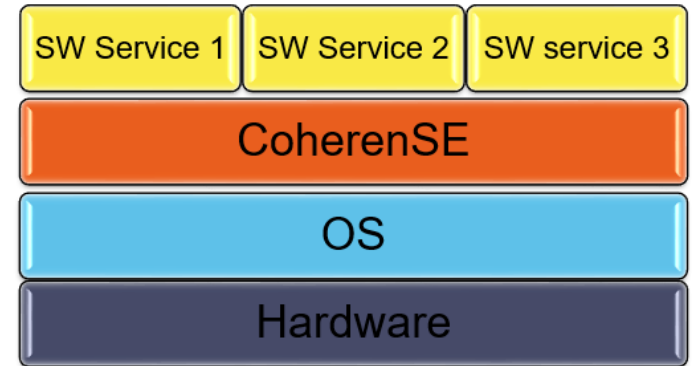
### Inconvénients de l'architecture traditionnelle

- Il existe plusieurs réseaux de véhicules différents
- Non optimisé et chaque calculateur nécessite un câblage
- Les réseaux connectés peuvent se chevaucher
- Difficile d'ajouter de nouveaux éléments (capteurs, calculateurs, ....)

### Cette architecture n'est pas adaptée aux véhicules de demain

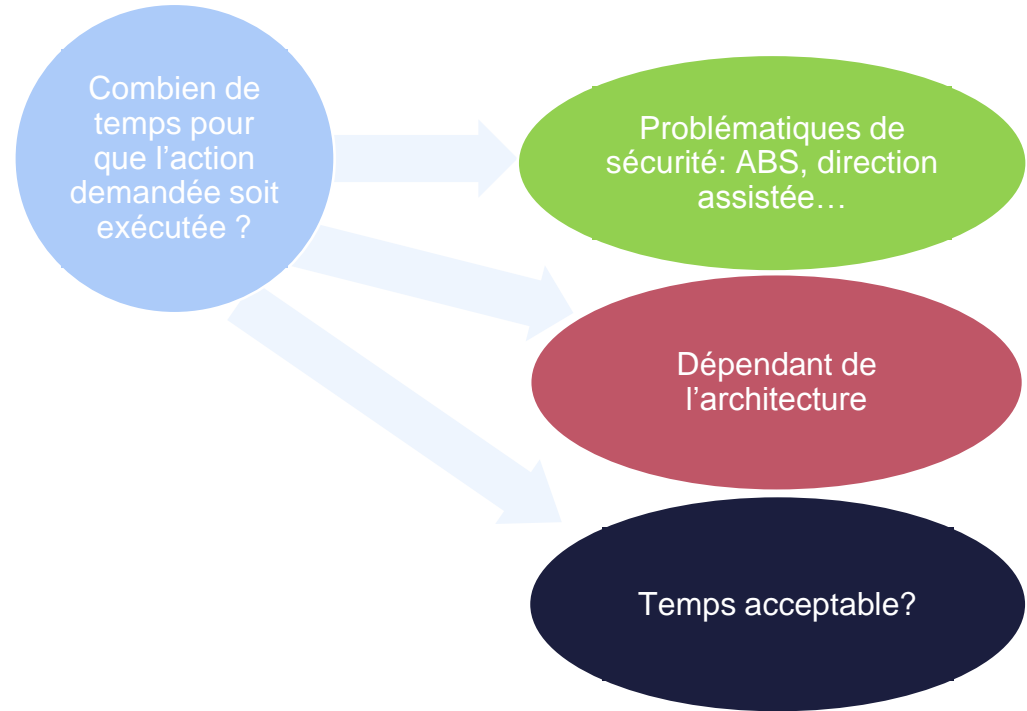
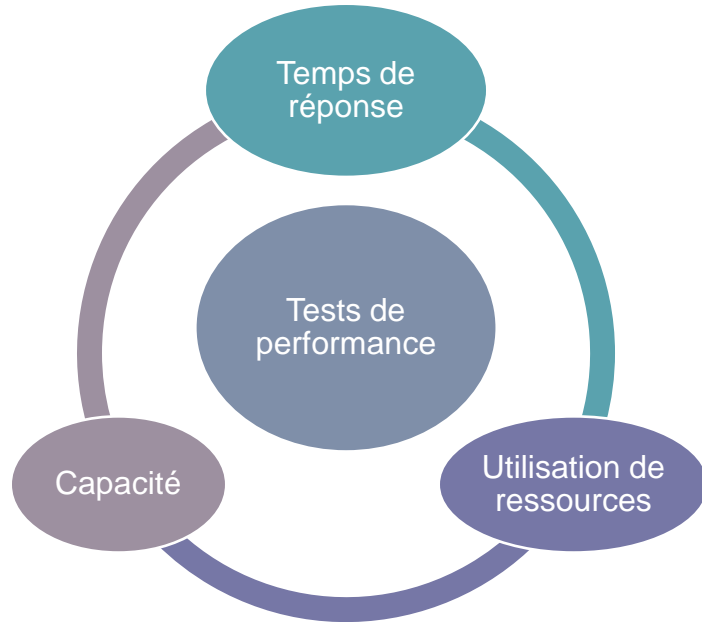
### Les nouvelles architectures : le projet CoherenSE

- + Simplifier/standardiser les architectures automobile en intégrant les principes de l'Architecture Orientée Services (ou SOA) au monde embarqué
- + Intégration/gestion simple des services (options)
- + Réduction du nombre de calculateurs

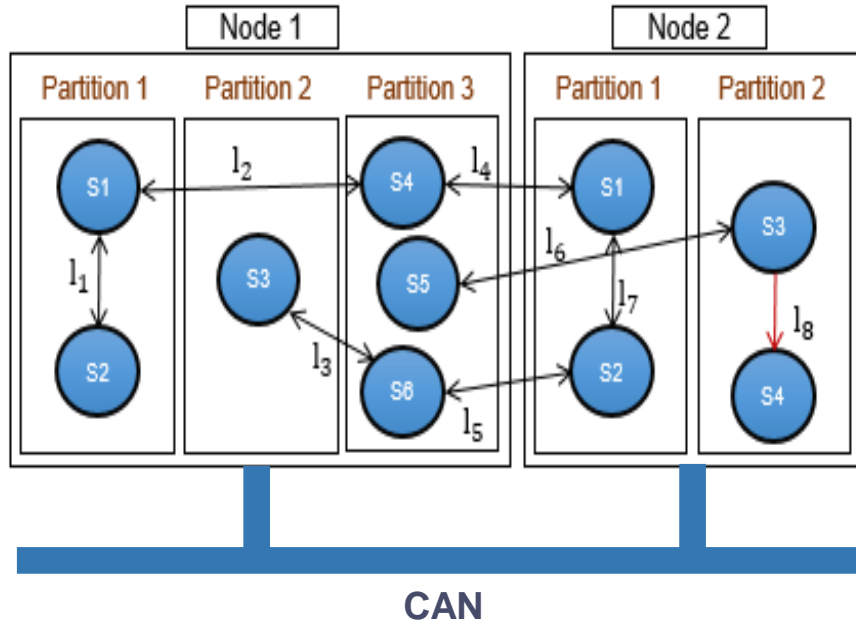


Co-développé avec Jaguar Land Rover

# Les tests de performance du système CoherenSE



# Architecture détaillée et communication



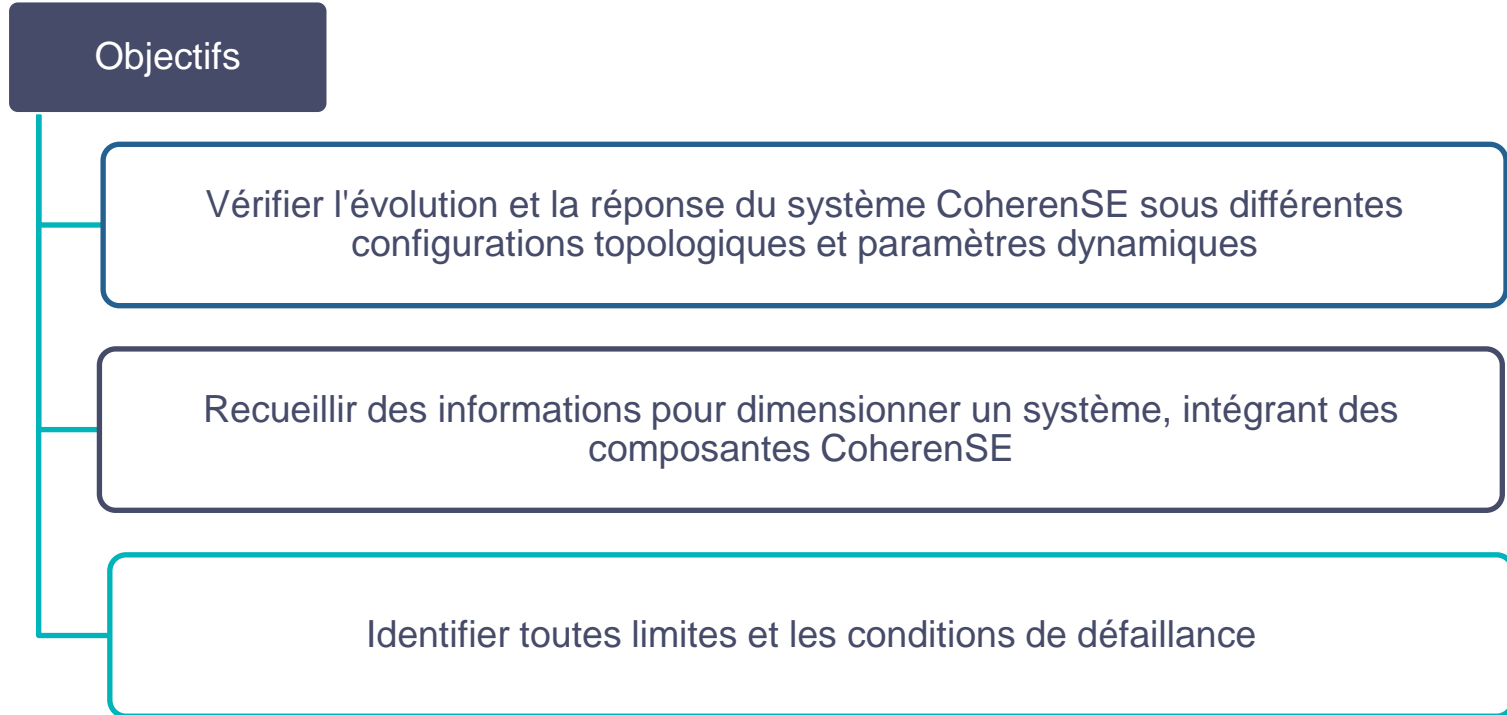
Toute architecture CoherentSE comporte trois parties fondamentales :

- **Nœud** : Unité de traitement informatique
- **Partition** : Environnement d'exécution
- **Service** : La plus petite fonctionnalité du système

La communication entre services peut être:

- **Intra-partition** (Ex :  $l_1$ ) ,
- **Inter-partition** (Ex :  $l_2$ ) ,
- **Inter-node** (Ex :  $l_6$ )

# Objectifs des tests de performance de CoherenSE



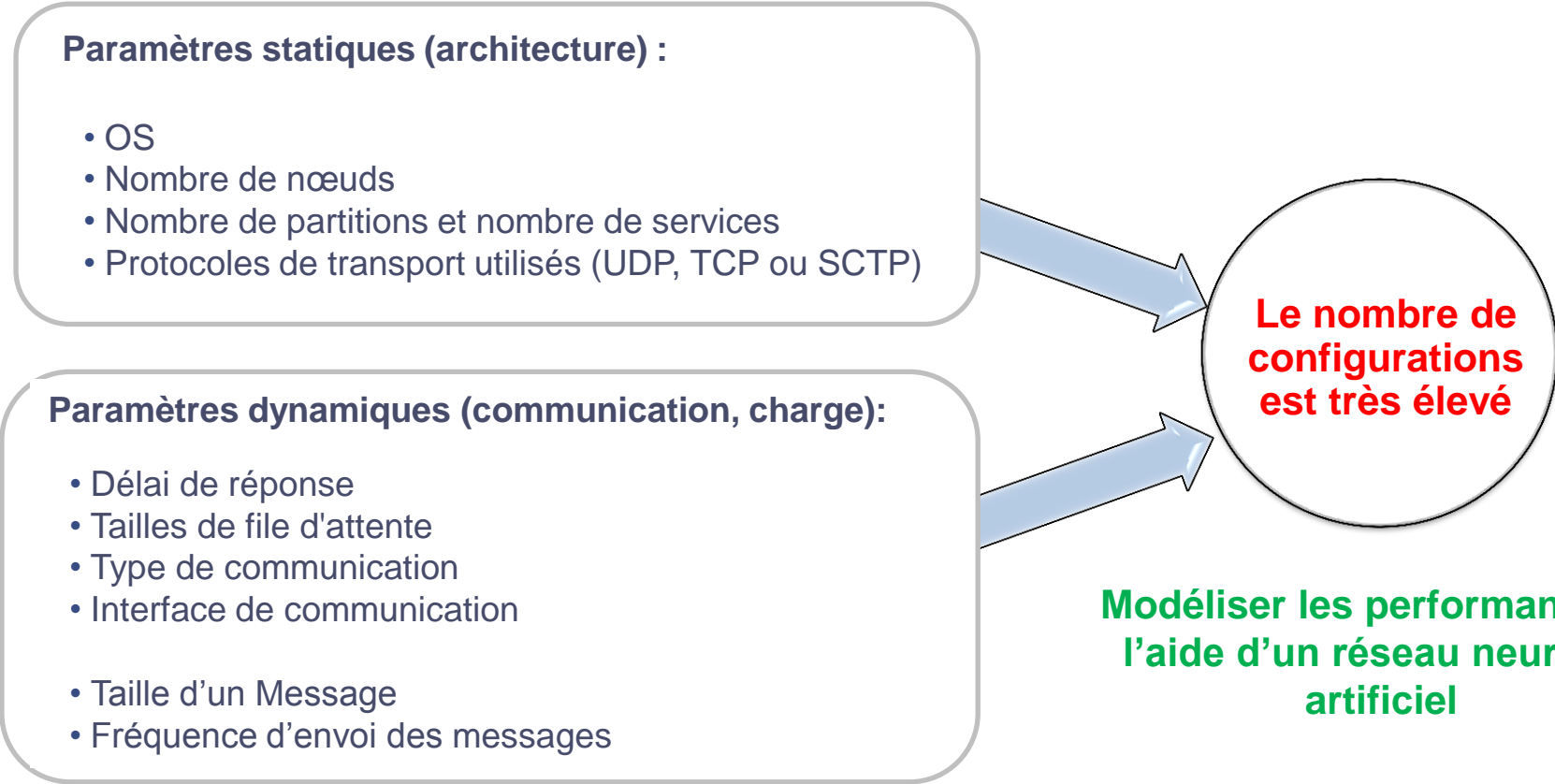
# Paramètres de configuration du système

## Paramètres statiques (architecture) :

- OS
- Nombre de nœuds
- Nombre de partitions et nombre de services
- Protocoles de transport utilisés (UDP, TCP ou SCTP)

## Paramètres dynamiques (communication, charge):

- Délai de réponse
- Tailles de file d'attente
- Type de communication
- Interface de communication
  
- Taille d'un Message
- Fréquence d'envoi des messages



**Le nombre de configurations est très élevé**

**Modéliser les performances à l'aide d'un réseau neuronal artificiel**



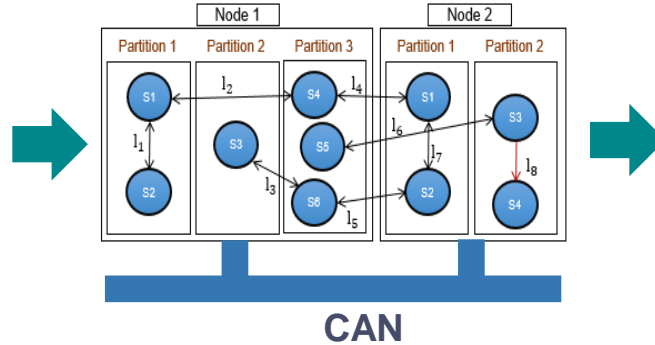
# 2.

Tests de performance et prédiction des performances par réseau de neurones

# Collecte de mesures de performance de référence

## Données en entrée

- Description de la topologie
- Paramètres dynamiques



## Données en sortie (performances)

- Temps de réponse
- CPU utilisé
- Mémoire utilisée
- Bande passante utilisée

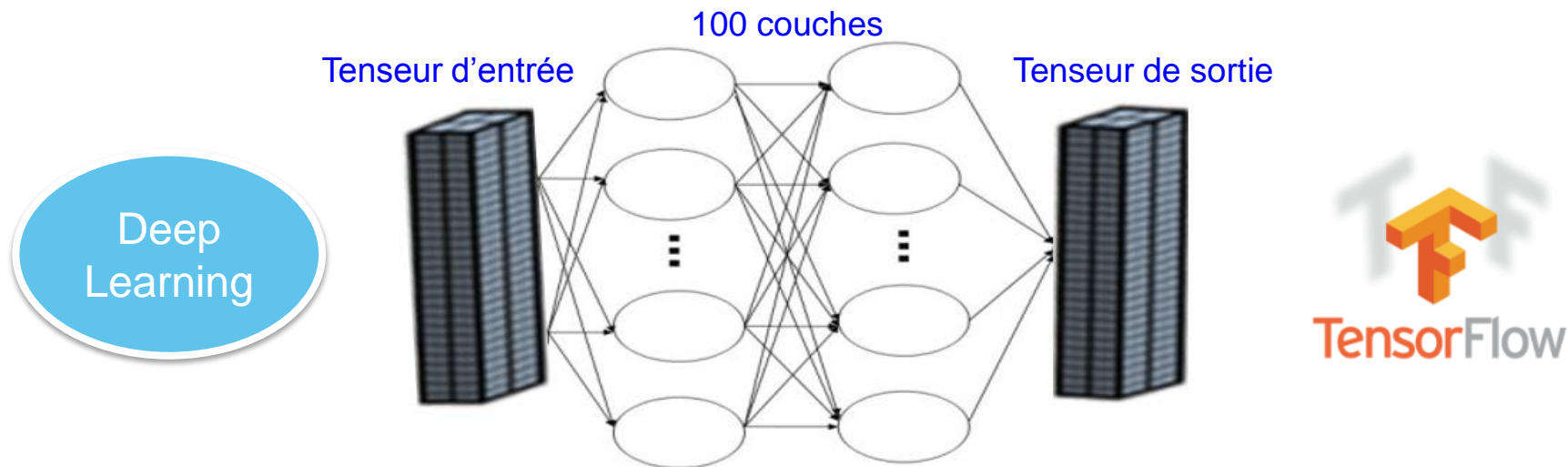
## De multiples configurations ont été testées

- Environ 10 architectures
- Plusieurs dizaines de configurations des paramètres dynamiques (tailles de message, trafic, files d'attente, ....)

## Elaboration d'un tableau de données en entrée et résultats des tests de performance

# Principe de prédiction des performances par réseau de neurones

Modéliser les paramètres de toute configuration par un tenseur d'entrée (caractéristiques du système à tester et charge appliquée) et la sortie (performances du système) par un tenseur de sortie



Le réseau de neurones est généré automatiquement (« entraîné ») par TensorFlow sur la base de l'ensemble des données d'entrée et de sortie

Il est ensuite capable de déterminer automatiquement les valeurs de sortie à partir des valeurs d'entrée

# Démarche de mise en œuvre de la prédiction des performances

1

## Collecte des données de référence

- Collection des performances de différentes configurations
- Classification des performances obtenues dans une base de données
- **L'entraînement du réseau de neurones nécessite des milliers de données**



2

## Prédiction des performances

- Prédiction des performances par le processus Deep Learning
- Entraînement du réseau neuronal artificiel profond
- Exécution du réseau neuronal et validation des prédictions

1



Dans un premier temps enrichissement de la base de données par **interpolation** (l'interpolation nécessite beaucoup moins de données que l'entraînement du réseau de neurones)



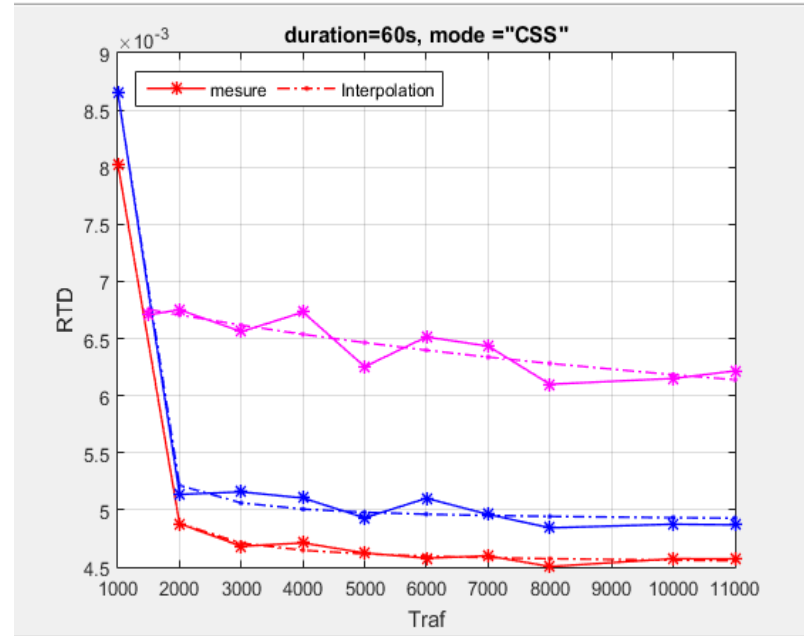
Ultérieurement **exécution automatique de tests de performance** pour enrichir la base de données

# Interpolation de mesures par méthodes mathématiques

Utilisation de MatLab pour évaluer les corrélations entre les variables (taille des messages, trafic,...)

Identification du modèle mathématique utilisé pour générer le jeu de données:

$$RTD(\text{traffic}, \text{Msg\_size}) = \frac{x_1(\text{link}_{m,\text{type}}) \text{Msg\_size}}{\text{traffic} + x_2(\text{link}_{m,\text{type}})} + x_3(\text{link}_{m,\text{type}})$$

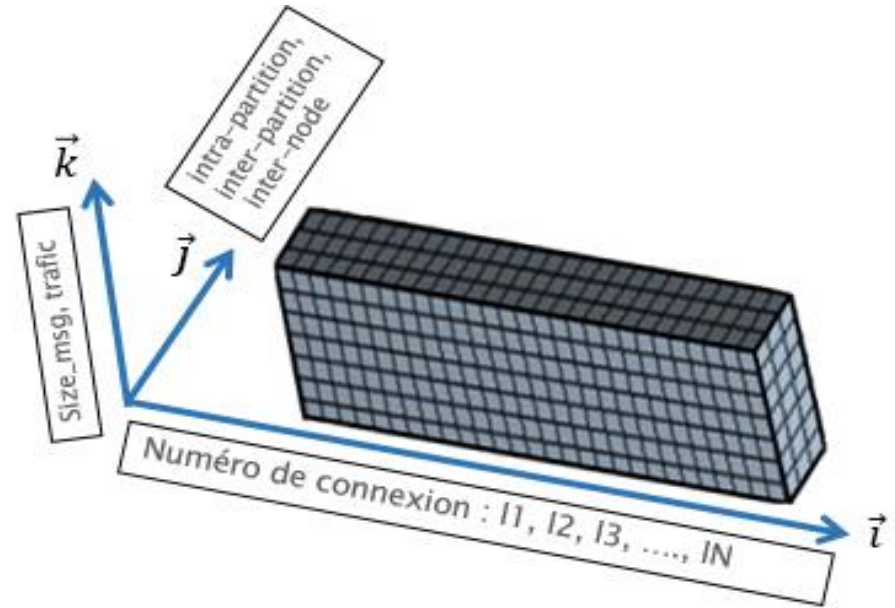


# Entraînement du réseau de neurones – données en entrée et sortie

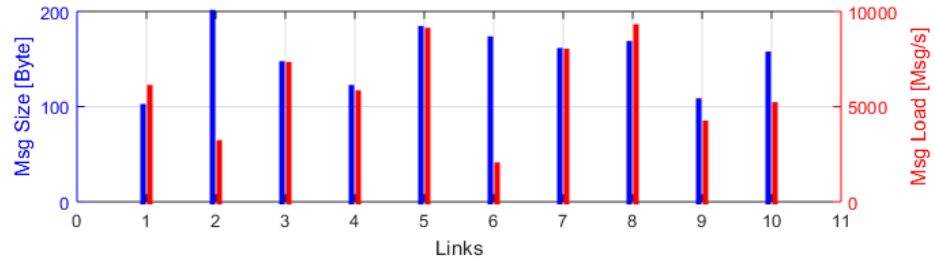
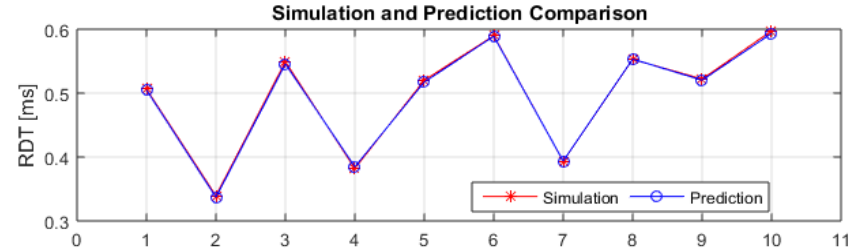
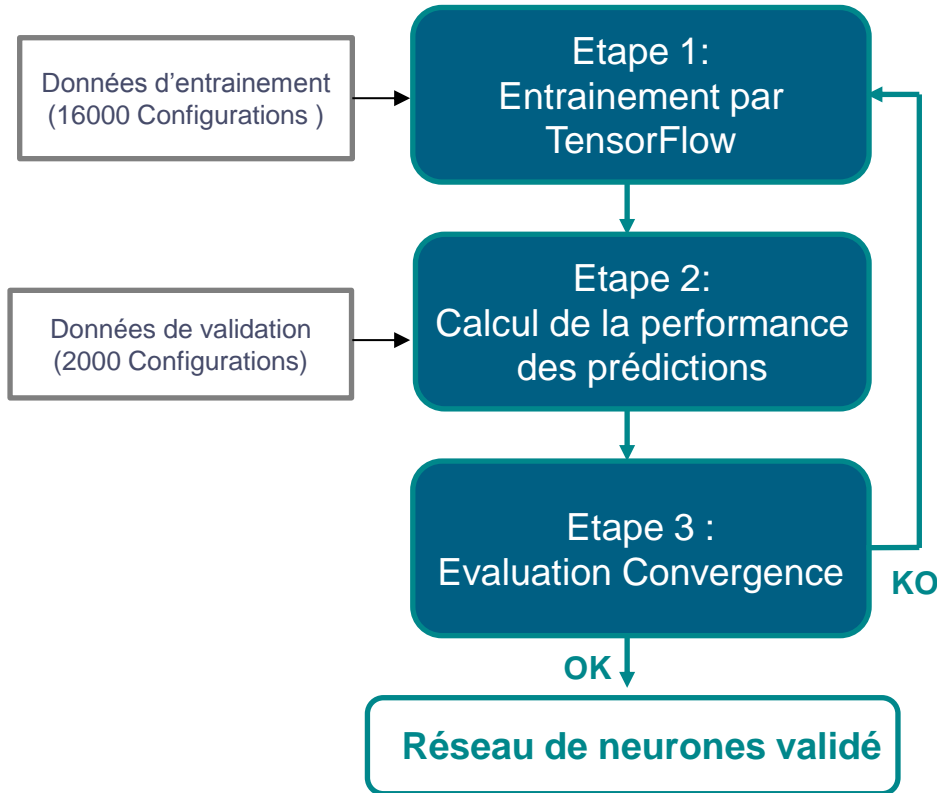
➔ Une configuration d'entrée est modélisée par un tenseur tridimensionnel :

- ✓ Dimension 1 (axe  $i$ ) : les numéros de communication  $l_1, l_2, \dots, l_N$ .
- ✓ Dimension 2 (axe  $j$ ) : les types de communications.
- ✓ Dimension 3 (axe  $k$ ) : les propriétés des communications et la charge qui leur est appliquée.

➔ Les données de sorties sont modélisées par les performances du système (à ce stade limitées aux temps de réponse)



# Validation des prédictions du réseau de neurones



Précision de convergence du modèle : **97 %**

# 3.

## Conclusion et perspectives



## Tests de performance CoherenSE : La suite

Enrichissement de la base de mesures de références à l'aide de tests automatisés afin d'améliorer la précision du modèle.

Implémentation d'une nouvelle architecture de réseau de neurones capable de prédire à la fois :

- ✓ Temps de réponse
- Consommation CPU
- Consommation de mémoire
- Bande passante (capacité)

# Conclusion

- Les prédictions de performance sont très proches des mesures effectuées sur des prototypes.
- La détermination de la performance d'une configuration est instantanée et ne nécessite pas d'environnement.
- Il est ainsi possible d'évaluer automatiquement la performance d'un très grand nombre de configurations, afin d'identifier les architectures optimales.
- Les démarches mises en œuvre peuvent s'appliquer à d'autres technologies (SOA, potentiellement tout type de tests de performance):
  - Interpolation de mesures de performance
  - Prédiction des performances par réseau de neurone (nécessite un plus grand nombre de données)

# Questions