

Développement d'une plateforme de simulation système-réseau

Rémi Goalard - 4AE-SE. Tutrice : Oana Hotescu

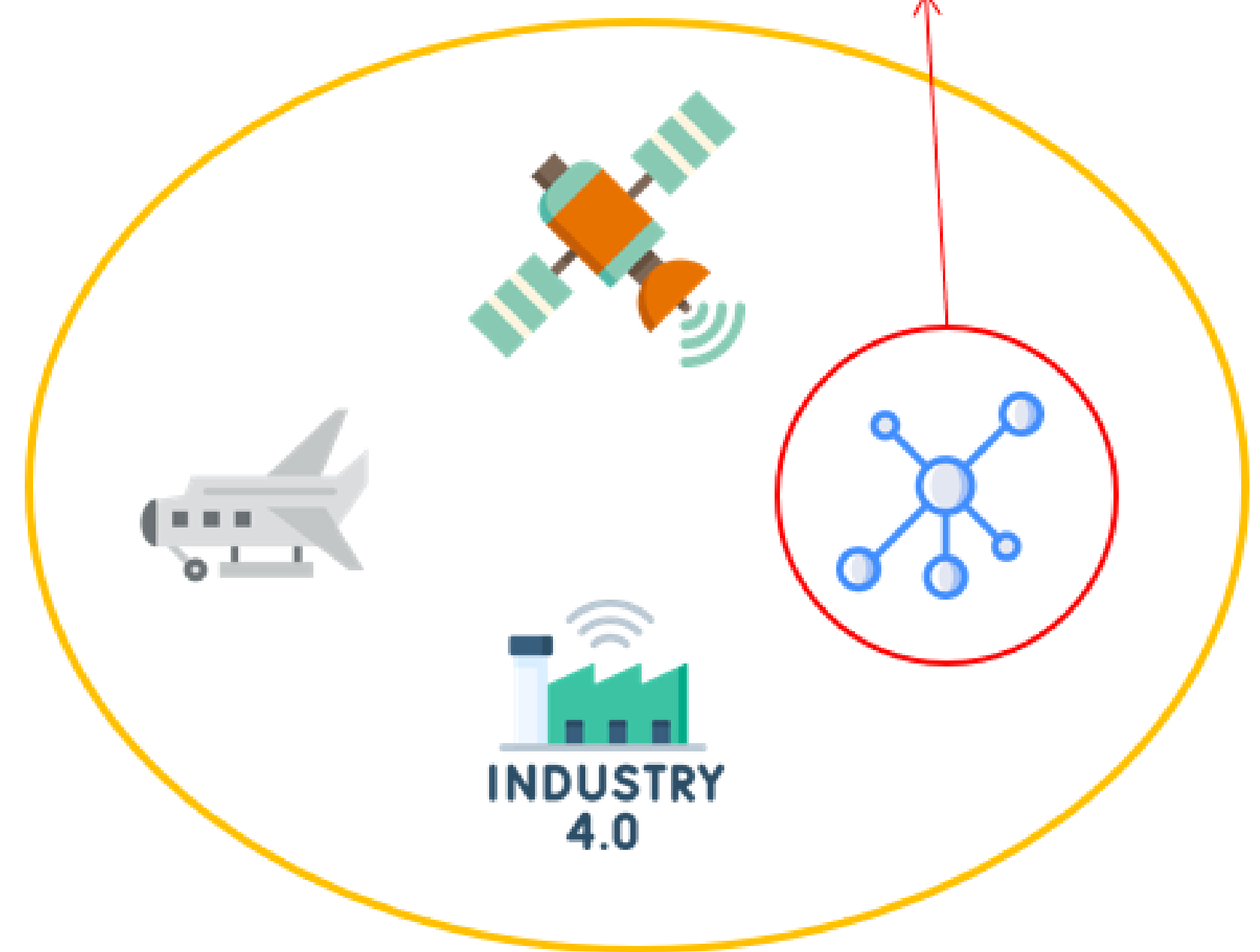
ISAE SUPAERO - Institut Supérieur de l'Aéronautique et de l'Espace

Contexte

- La grande majorité des domaines de l'industrie contiennent des systèmes dits temps-réel. Il s'agit de systèmes dont le respect des contraintes temporelles est aussi important que la validité des résultats.
- Dans le cadre de l'étude des réseaux, les systèmes temps-réels sont appelés TSN (Time Sensitive Networks). Ils sont définis par la norme IEEE 802.3.
- Il est possible de simuler des topologies de TSN via des logiciels comme OMNET++ afin d'étudier les temps de réponses sur ces réseaux, mais aucun ne permet d'avoir d'informations sur la consommation énergétique des machines présentes sur le réseau. L'objectif est d'ajouter la dimension '*consommation énergétique*' sur OMNET++.

TSN : IEEE 802 :

- Réplication des frames
- Elimination des frames
- Synchronisation
- QoS



Contribution

- Réflexion sur l'introduction de la '*consommation énergétique*' sous OMNET++, 2 pistes :
 - Développer un module *CPU* sous OMNET++. Le module associe une consommation énergétique statique à chaque message émis ou reçu sur un noeud du réseau.
Problème : L'énergie consommée par un CPU n'est pas statique, dépend de facteurs non considérés :
 - 1) Chaque message reçu sur un noeud doit être traité -> change l'ordonnancement du CPU
 - 2) Mécanismes d'optimisation existant sur les CPU réels : DVFS, Little/big...
 - Simuler le comportement d'une architecture CPU face à un taskset (hors OMNET++), et utiliser l'ordonnancement produit pour émettre des messages sur un réseau OMNET++.
Problème : Besoin d'un logiciel simulant le comportement d'un CPU avec mécanismes d'optimisation, qui communique en temps réel avec OMNET++ afin de traiter les messages reçus sur le réseau.
- Recherche et prise en main sur le logiciel expérimental RTSim (Real Time Simulator), émulateur d'architectures CPU. RTSim fournit l'ordonnancement et la consommation associés à un taskset et une architecture donnés. Création d'un script 'bash' pour récupérer les données de sortie et les ranger dans un fichier '.xml'.
- Réflexion sur l'intégration de RTSim dans OMNET++ <=> communication entre les noyaux des deux simulateurs.
- Travail de documentation du code source RTSim, pour faciliter son utilisation et son intégration sous OMNET++.

Résultats

- Découverte du framework **RTSim**, simulant une architecture CPU.
- Amélioration de la lisibilité/exploitabilité des données sortant de RTSim.
- **Echec** de l'intégration de RTSim dans OMNET++
- Début de la documentation de RTSim.

Conclusions

Echec de l'intégration de RTSim sous OMNET++. En revanche, la documentation commencée pourrait servir à comprendre la façon dont est construit RTSim, et s'en inspirer pour reproduire sous OMNET++ les fonctionnalités intéressantes du framework.