

TUTEUR(S)

- Jerome GUEUNIER jerome.gueunier@actia.fr
- Guillaume Auriol <g_auriol@insa-toulouse.fr>
- Slim Abdellatif <slim@laas.fr>

TITRE DU PROJET

Communication Ethernet déterministe sous Linux

(Orchestrateur Time-Triggered, sockets dédiées et cohabitation avec flux TCP/IP)

MOT-CLES

Transmissions déterministes, Linux, programmation système, réseaux

DESCRIPTIF (RESUME)

Les systèmes embarqués distribués échangent de plus en plus de données sur Ethernet. Certaines données peuvent être transmises de manière classique via TCP/IP, tandis que d'autres doivent être envoyées à des instants précis et prévisibles. L'enjeu du projet est d'explorer une communication Ethernet planifiée, dite time-triggered, dans un environnement Linux simple.

ACTIA souhaite étudier une preuve de concept composée de deux nœuds Ethernet. Le premier nœud joue le rôle d'orchestrateur et pilote le déroulement d'un cycle de communication. Le second nœud participe au réseau et échange des messages périodiques avec l'orchestrateur. L'objectif est de faire cohabiter des messages time-triggered, envoyés dans des créneaux temporels fixes, avec des flux TCP/IP classiques.

Principe de fonctionnement visé

Le démonstrateur visera un cycle de communication périodique de 100 ms, découpé en créneaux temporels de 10 ms. Les premiers créneaux seront réservés aux échanges time-triggered entre les deux nœuds, tandis que certains créneaux suivants pourront être utilisés par des flux TCP/IP. Un dernier créneau pourra être conservé pour l'attente, la resynchronisation ou la préparation du cycle suivant.

À titre de scénario initial : le TSlot_0 correspondra à un message de l'orchestrateur vers le second nœud et pourra servir de message de synchronisation. Le TSlot_1 correspondra à un message du second nœud vers l'orchestrateur. Les TSlots suivants pourront accueillir du trafic TCP/IP ou des flux de charge destinés à perturber le réseau.

Problématique

Comment mettre en place sous Linux une première maquette Ethernet permettant de transmettre certains messages dans des créneaux temporels planifiés, tout en laissant cohabiter des communications TCP/IP classiques ?

Objectif du PIR

L'objectif du PIR est de réaliser une étude exploratoire et une maquette logicielle permettant de valider les principes d'un orchestrateur time-triggered placé entre les couches protocolaires et l'interface réseau. Le projet ne vise pas une solution industrielle complète ni une modification lourde du noyau Linux. Il doit produire une preuve de concept compréhensible, mesurable et documentée, qui permettra à ACTIA de décider des suites possibles.

Travail demandé

- **État de l'art ciblé** : Étudier les notions nécessaires relative à « linux networking » et les principes des communications time-triggered (guidées par le temps).
- **Architecture de la maquette** : Définir une architecture à deux nœuds : un orchestrateur et un membre du réseau. Décrire le rôle de chaque nœud, le cycle de communication, le nombre de time slots et la fonction de chaque créneau.
- **Plan de communication** : Proposer un format simple de plan de communication, par exemple en JSON, contenant le rôle du nœud, la durée du cycle, le nombre de slots, les fenêtres TX/RX et les adresses MAC associées.
- **Prototype logiciel** : Développer une première couche logicielle d'orchestration en espace utilisateur. Cette couche devra ouvrir ou autoriser l'émission/réception de trames dans les créneaux prévus.
- **Socket time-triggered simplifiée** : Étudier la faisabilité d'une interface de type socket time-triggered. Pour un PIR, une abstraction logicielle ou une bibliothèque utilisateur peut suffire si une vraie extension noyau est trop ambitieuse.
- **Daemon de configuration** : Réaliser, si possible, un petit daemon ou script de configuration chargé de lire le plan de communication JSON et de configurer le comportement du nœud au démarrage.
- **Cohabitation avec TCP/IP** : Générer des flux TCP/IP classiques pendant certains créneaux et observer leur impact sur les messages time-triggered.
- **Mesures et analyse** : Mesurer les écarts temporels, la latence, la gigue et les pertes éventuelles. Comparer le comportement avec et sans orchestration temporelle.

Livrables attendus

- Rapport d'état de l'art synthétique sur Ethernet, TCP/IP, adressage MAC, sockets Linux et communications time-triggered.
- Spécification simplifiée de la maquette : rôles des nœuds, cycle, time slots et plan de communication.
- Format JSON documenté pour décrire le plan de communication.
- Prototype logiciel ou scripts permettant de piloter les échanges time-triggered sur deux nœuds Linux.
- Démonstration de cohabitation entre messages planifiés et flux TCP/IP.
- Résultats de mesure : latence, gigue, respect des créneaux, limites observées.
- Recommandations pour une suite : passage éventuel en noyau, usage de mécanismes Linux avancés, besoin matériel, précision temporelle et perspectives TSN.

Niveau d'ambition adapté à un PIR

Le projet doit rester exploratoire : comprendre, maquetter, mesurer et documenter. Il n'est pas attendu de développer un protocole Ethernet industriel complet. Il n'est pas obligatoire de modifier le noyau Linux si une solution en espace utilisateur permet de démontrer les principes. La précision temporelle absolue n'est pas l'objectif principal ; l'objectif est d'identifier les mécanismes, les gains et les limites d'une approche time-triggered sous Linux. La maquette doit être reproductible par ACTIA à partir des scripts, de la configuration et du rapport fournis.

Compétences développées

- Réseaux Ethernet et TCP/IP
- Programmation système sous Linux
- Composante communication du système Linux
- Sockets et communication bas niveau
- Ordonnancement temporel et mesure de latence
- Configuration réseau et scripts Python/systemd
- Expérimentation, analyse critique et rédaction technique

Pourquoi choisir ce sujet ?

Ce sujet permet aux étudiants de travailler sur une problématique très concrète des systèmes embarqués distribués : comment utiliser une technologie standard comme Ethernet tout en maîtrisant mieux le moment où certaines données critiques sont transmises. Le projet combine réseau, Linux, programmation système, expérimentation et réflexion d'architecture.

Il est particulièrement adapté à un PIR car il propose une vraie montée en compétence : partir des bases Ethernet/TCP-IP, construire une maquette simple, observer des phénomènes mesurables, puis formuler des recommandations utiles pour une suite industrielle.

Remarque : Le périmètre final sera ajusté avec l'encadrement ACTIA/INSA. Une implémentation en espace utilisateur, basée sur scripts Python, sockets et configuration systemd, est considérée comme suffisante pour une première preuve de concept PIR.

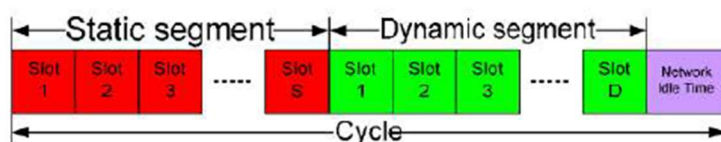
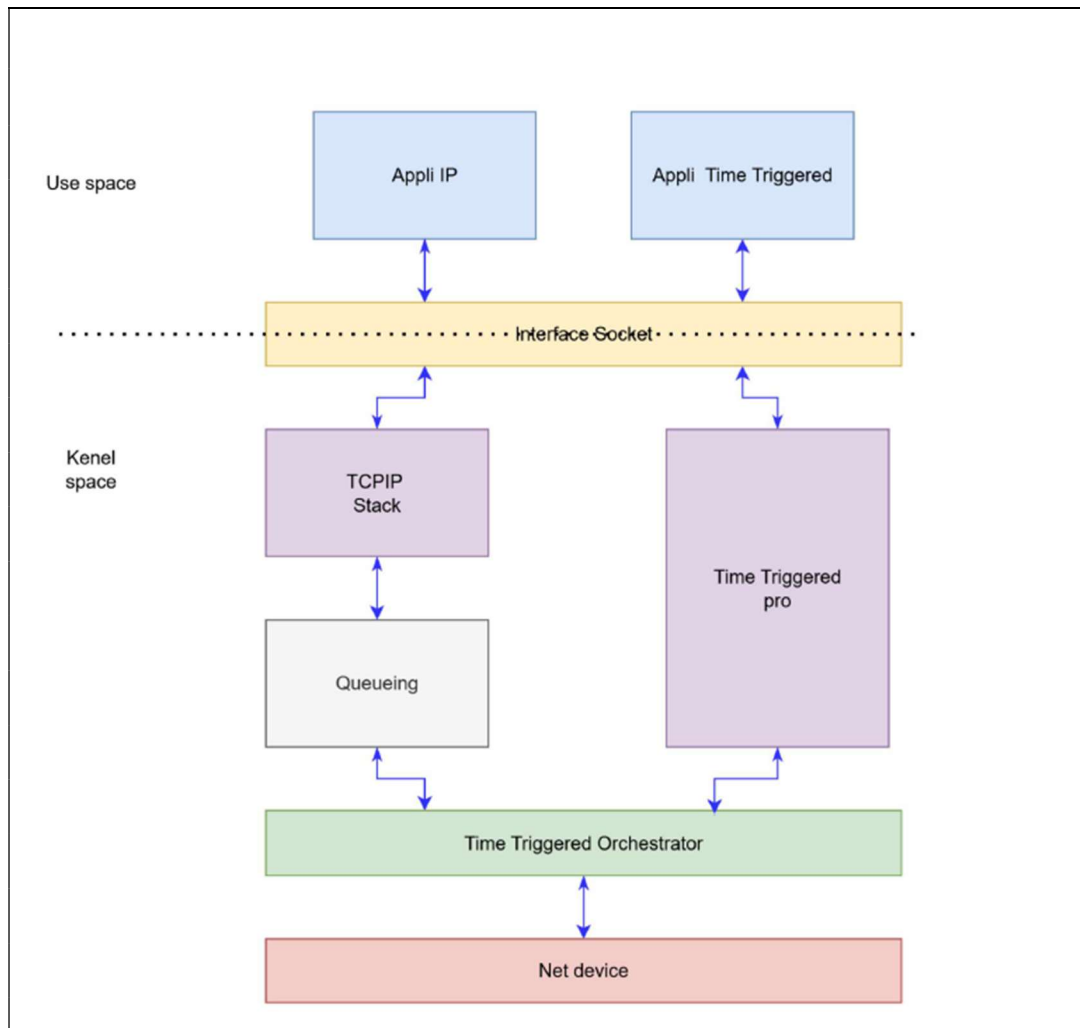


Figure 2: A communication cycle with both event- and time-triggered slots.



PROFIL DES ETUDIANTS SOUHAITE (1 seul choix par projet)

- ☒ AE-SE : spécialité Automatique-Electronique parcours Systèmes Embarqués
- ☒ IR-SI : spécialité Informatique parcours Systèmes Informatiques
- ☒ IR-SC : spécialité Informatique parcours Systèmes Communicants

☐ (optionnel) ce projet peut être proposé à un ou des étudiants d'échange sur la partie réalisation seule (semestre 1 et/ou semestre 2)

PRIORITE : si vous posez plusieurs sujets, indiquer ici la priorité de ce sujet (1= plus prioritaire)