

Etude de protocoles et modélisation de réseaux Space Wire et Space Fibre pour application satellitaire

Rapport d'expérience stage de césure
Mars - Août 2017

Ludovic Thomas
Tuteur : Brice Dellandréa
Tutrice école : Ahlem Mifdaoui

Introduction

J'ai réalisé la seconde partie de mon stage au département de recherche et développement plateforme et avionique à *Thales Alenia Space* à Cannes. Celui-ci portait d'une part sur l'analyse des protocoles pour SpaceWire et SpaceFibre et d'autre part sur le développement d'un outil de modélisation pour l'étude de réseaux utilisant ces protocoles.

Contexte de l'étude

L'entreprise Thales Alenia Space

Thales est une entreprise parmi les leader sur les systèmes critiques pour les transports, la sécurité et l'aérospatial, détenue à 25.8% par l'État français. Créé en 2000, le groupe bénéficie en réalité de l'expertise accumulée sur plus d'un siècle au sein de nombreuses entreprises historiques parmi lesquelles on peut citer Thomson-CSF, Alcatel Space, SGS-Thomson (qui deviendra STMicroelectronics) ou Sextant Avionique. Le groupe est actuellement fort de soixante quatre mille salariés répartis sur cinquante-six pays et est connu pour sa forte dynamique d'innovation et la valorisation duale (activités civiles et militaires) de ses compétences techniques. C'est avec Leonardo, une entreprise italienne spécialisée également sur les transports, l'aérospatial et les systèmes de défense que Thales possède la co-entreprise Thales Alenia Space (TAS). Cette dernière forme, avec l'entreprise italienne *Telespazio* la *Space Alliance* qui propose un large éventail de solutions orbitales adaptées au marché. Parmi les activités de TAS, on peut noter les télécommunications (constructeur de la constellation Iridium Next), l'observation de la terre (maître d'œuvre de 17 satellites Météosat), l'exploration (constructeur du Trace Gaz Orbiter), la navigation (maître d'œuvre du système EGNOS) et les infrastructures orbitales (constructeur notamment de la Cupola pour l'ISS).

Le département R&D, avionique et plateforme

Le département de recherche et développement, avionique et plateforme de l'établissement Cannois accomplit des activités transverses de recherche et développement de technologies (R&T) et réalise avant-projets et études de faisabilité dans des domaines variés comme les systèmes de traitement de données, le contrôle d'attitude et d'orbite, le traitement d'image et la distribution de puissance. L'un des enjeux majeurs réside dans la veille et la réactivité aux orientations politiques spatiales en Europe et dans le monde afin de rester un leader technologique incontournable lors de l'application des politiques. Ces activités nécessitent souvent des collaborations dans le cadre de groupes de travail rassemblant industriels, agences spatiales et laboratoires de recherche. Celles-ci permettent à *Thales Alenia Space* d'être force de proposition dans les orientations technologiques au niveau européen et mondial. C'est ainsi qu'on peut noter par exemple la présence de TAS comme membre moteur dans les groupes de travail sur le SpaceWire ou le SpaceFibre.

Les activités DHS

Les systèmes de traitement des données embarqués (DHS, On-Board Data Handling Systems) constituent le cœur de l'avionique d'un système spatial chargé de la récolte, de l'agrégation, du traitement et du transport de l'information. Les différentes communications transitant sur le réseau d'un système spatial présentent une diversité de besoin en débit, latence (temps entre l'émission et la réception), *jitter* (écart-type de la latence) et en garanties. L'analyse fonctionnelle cherche donc à classer ces communications selon leurs besoins.

Le besoin en QoS, Quality of Service désigne la garantie de réception nécessaire à une communication sur un réseau. Ce besoin est classé en trois niveaux :

- 0 - Best effort. La communication ne nécessite aucune garantie. L'émetteur du message procédera à son transfert une unique fois, sans garantie d'arrivée au destinataire.

- 1 - At least one. Ici, le destinataire de l'information a besoin de recevoir le message au moins une fois.
- 2 - Exactly once. Dans ce dernier niveau, le destinataire doit recevoir une et une seule fois le message. Cette garantie est typiquement nécessaire lorsque la communication concerne le contrôle du satellite.

Le niveau de déterminisme d'un réseau désigne un ensemble de garanties sur la latence et sur sa variation aléatoire lorsque l'information est effectivement transmise de l'émetteur au destinataire.

- Non déterministe. Aucune garantie n'est apportée sur le temps de propagation de l'information.
- Temps de livraison garantie, ou *Soft determinism*. Ici, le temps de propagation d'un émetteur à son destinataire est garantie par une valeur maximale. On fixe donc le maximum de la latence mais pas le jitter.
- Déterministe. Ici, latence et *jitter* sont fixés sous la forme d'une fenêtre.

La QoS peut se combiner avec n'importe quel niveau de déterminisme selon les besoins de la communication. Par exemple, une communication nécessitant une QoS 0 et un déterminisme ne nécessite pas de garantie de réception, mais en cas de réception, la latence doit se situer dans la fenêtre fixée. Enfin, certains transferts de données peuvent également présenter des besoins du point de vue de la détection, l'identification et la résolution des erreurs (FDIR, *Failure Detection, Failure Identification and Recovery*). En effet toute erreur pouvant intervenir sur le réseau peut se classer parmi l'un des niveaux suivants, en fonction de la réponse que peut apporter le sous-système DHS à cette erreur :

- Niveau 0 : L'erreur est détectée, identifiée et corrigée par le sous-système.
- Niveau 1 : L'erreur est détectée et identifiée par le sous-système, mais celui-ci ne peut la corriger localement.
- Niveaux 2 et supérieurs : L'erreur n'est pas identifiée directement mais sa présence est détectée par des incohérences relevées au niveau sous-système (temps de propagation en augmentation, débits incohérents...), du système satellitaire global (incohérences de tension, d'attitude), ou au niveau logiciel (erreur *software*).

Il faut noter que toute erreur d'un niveau supérieur à 1 est déportée au système supérieur, et pour toute erreur d'un niveau supérieur à 2, celui-ci ne dispose pas des éléments permettant de localiser la source du défaut. En fonction de l'importance du contenu des communications, ces dernières peuvent donc aussi imposer les niveaux de résolution (généralement, le plus localement possible, donc 0 ou 1) de certains types d'aberrations prévisibles afin de

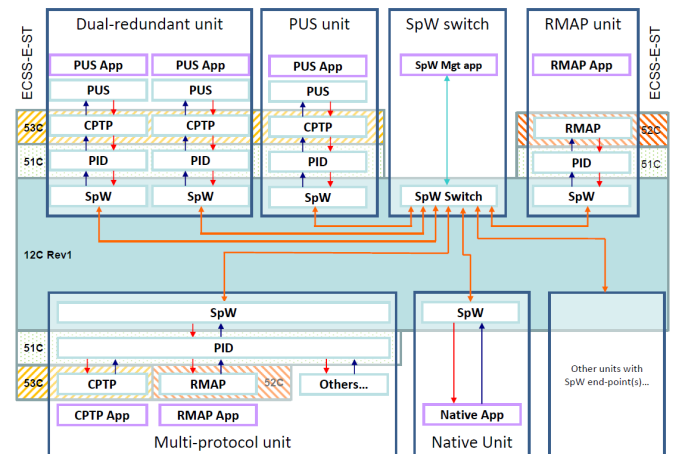


Figure 1: Exemple d'architecture SpaceWire avec différentes piles

limiter le temps nécessaire à la restauration du fonctionnement nominal. L'étude fonctionnelle consiste donc à associer les différents besoins (bande passante, qualité de service, fréquence des messages, *jitter* et latence moyens, niveau de déterminisme) pour obtenir des classes de communication et définir des principes et des exigences de base pour tout système DHS. C'est l'esprit notamment du projet OSRA-NET (*On-Board Reference Architecture*) auquel participe TAS. Par ailleurs, l'interconnexion de ces classes est vu comme l'opportunité de gagner du temps de développement des systèmes spatiaux, de la masse et de permettre l'association de métadonnées systèmes aux données scientifiques.

SpaceWire et la pile PID

Pour répondre à ces exigences, SpaceWire est une technologie de réseau point à point spécifiquement conçue pour les applications spatiales par un groupe de travail dans lequel *Thales Alenia Space* est un des principaux moteurs. SpaceWire est utilisé par de nombreuses missions spatiales parmi lesquelles ExoMars, Rosetta (ESA), le James Webb Space Telescope (NASA), BepiColumbo (JAXA). Il est en constante amélioration et sa première révision est en préparation. Il ne couvre lui-même (standard ECSS-E-ST-50-12C) que les couches "matérielles" du modèle OSI, avec une structure en paquets, un système de contrôle de flux et des choix orientés sur la performance et l'économie d'énergie (utilisation de la technologie *wormhole* pour le routage, utilisation de la technologie LVDS, *Low Voltage Data Signaling* pour un fonctionnement à faible puissance). Cependant il rend la coexistence de protocoles différents possible grâce à l'utilisation d'un *Protocol Identifier* (série des standards ECSS-E-ST-50-5X). Ceux-ci couvrent généralement la couche "Transport" du modèle OSI et supportent chacun une ou plusieurs classes de communication. La figure

Nom	QoS	Déterministe	Paquet détruit (Niv. FDIR)
CPTP	0	Non	>2
RMAP	0	Non	1 ou >2
SpW-D	0	Oui	1
STP-ISS	0 ou 2	Oui ou Non	0 ou 1

Figure 2: Tableau récapitulatif des caractéristiques principales des protocoles étudiés

1 présente ce principe de pile et montre l'une des nombreuses possibilités d'agencement que SpaceWire rend possible. Chaque protocole devant respecter les besoins liés à sa classe de communication, il est alors primordial de modéliser les comportements de la couche SpaceWire d'une part et des protocoles d'autre part afin de s'assurer de leurs performances et de l'absence de conflits. Les protocoles étudiés dans le cadre du développement de MOST sont :

- Le CCSDS (*Consultative Committee for Space Data Systems*) Packet Transfert Protocol (CPTP) est un protocole de transport de cargo d'une application à une autre, avec un *header* de taille minimale.
- Le protocole RMAP (*Remote Memory Access Protocol*) a été conçu plus particulièrement pour accéder à des registres et mémoires avec un minimum d'activité *software*.
- Le SpW-D, SpaceWire Déterministe, conçu sur RMAP, utilise un emploi du temps pour assurer le caractère déterministe des commandes RMAP.
- Le STP-ISS, qui possède un mécanisme de priorité ainsi qu'un emploi du temps.

La figure 2 résume certaines caractéristiques de ces différents protocoles. La dernière colonne indique le niveau FDIR de l'erreur potentiellement générée si un paquet est détruit lors de sa traversée du réseau. Lorsque le mot clé "ou" est présent, cela signifie que la caractéristique dépend de la configuration ou du type de service utilisé dans le protocole. Comme on peut l'observer, le protocole SpaceWire-D et STP-ISS ont des caractéristiques particulièrement proches.

Travaux réalisés

Développement de l'outil de simulation MOST

C'est dans cette optique que le projet MOST, *Modelling Of SpaceWire Traffic* est né. MOST est un simulateur de réseau SpaceWire qui permet :

- L'évaluation des performances d'un protocole, sa comparaison et sa coexistence avec les autres dans les toutes premières phases de définition de ce protocole.

- Une aide à la conception et aux choix des architectures réseaux pour des missions données.
- La vérification des performances d'une architecture (débits, utilisation des buffers, etc.), dans les phases précoces du projet et sans disposer des équipements réels.
- L'étude de l'impact sur le réseau global du non respect des spécifications par un fournisseur.
- L'étude des cas dégradés.

Le projet MOST a été lancé depuis plusieurs années et s'est initialement porté sur le framework OPNET comme environnement de simulation. L'implémentation de SpaceWire, des protocoles CPTP et RMAP et même de SpaceFibre lui ont permis de devenir le premier et actuellement unique simulateur de réseau SpaceWire entièrement fonctionnel. Cependant, la licence OPNET limitait le nombre d'acteurs du projet. C'est pourquoi mon stage rentre dans le cadre d'une étude de faisabilité de la portabilité de MOST sur le framework open-source Network Simulator 3. NS-3 est un simulateur de réseau par événements discrets¹ développé pour les usages Internet, principalement à destination de la recherche et de l'éducation. C'est un logiciel libre, sous licence GNU/GPLv2. Sa documentation explicite, sa stabilité et son caractère open-source en font un très bon choix pour développer MOST dessus. Néanmoins, certaines modifications du code source de NS-3 ont dû être apportées dans le respect de la licence pour rendre l'implémentation de MOST possible. Le développement de MOST sur NS-3 s'est déroulé en équipe de deux, j'ai pu travailler avec un apprenti possédant une expertise avancée de MOST sur OPNET, ce qui fut nécessaire bien que MOST sur NS-3 présente une structure sensiblement différente. Celle-ci est décrite en figure 3 et donne également l'organisation de notre travail :

- Le *SpaceWire model core* a été créé entre mars et mai, j'en avais principalement la charge. Un *tradeoff* sur la structure générale interne du core a été réalisé entre plusieurs options en fonction de leur flexibilité, de leur compatibilité avec NS-3 et de leur cohérence par rapport au standard SpaceWire. Finalement, la structure adoptée est découplée de NS-3, avec son propre *stack* et ses propres interfaces. L'usage massif des interfaces notamment au niveau caractère et de l'héritage² en fait un ensemble de quarante "briques" avec une bonne ségrégation et une très bonne modularité.
- Ces briques doivent ensuite être assemblées en structures cohérentes pour représenter de réels équipements. C'est le rôle des *helpers*, qui furent également à ma charge au cours du mois de juin 2017.

¹Un simulateur par événements discrets calcule les événements aussi vite que possible, chaque événement pouvant en générer d'autres dans le futur. Sauf exception, la simulation n'est pas en temps réel

²Notamment de l'héritage virtuel, c'est ce qui a nécessité certaines interventions sur le code source de NS-3, car ce dernier n'était pas conçu pour le gérer

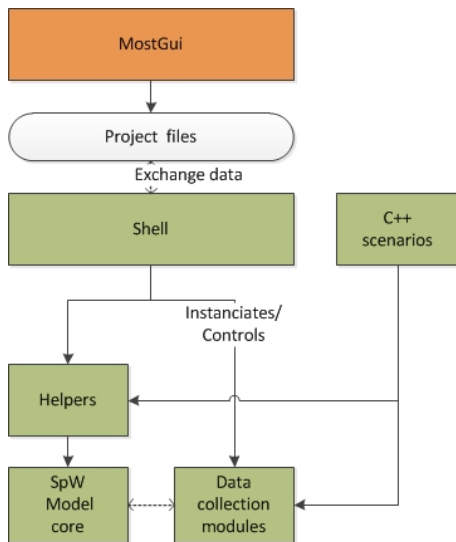


Figure 3: Structure du simulateur MOST

- Les *data collection modules* sont des unités C++ chargées de récolter les données générées par le *model core* pour en créer des graphiques exploitables. Ici aussi, après une réflexion à laquelle j'ai participé, le design retenu est particulièrement modulaire et laisse une flexibilité élevée aux utilisateurs de MOST. C'est mon binôme qui s'est chargé de leur implémentation à partir de mai.
- A partir de cet état et en utilisant l'API mis à disposition, un utilisateur externe programmant en C++ peut d'hors et déjà utiliser MOST avec des scénarios C++.
- Cependant pour donner également accès pour des utilisateurs ne programmant pas, une interface graphique basée sur le framework Qt5 fut développée. Elle est totalement fonctionnelle et échange un grand nombre d'informations directement avec le simulateur MOST pour s'adapter au modèle (l'ajout d'un composant, d'un protocole ou d'une fonctionnalité au simulateur MOST ne nécessite pas de modification de MostGui). Je fut en charge de la partie MostGui tandis que mon binôme fut en charge du Shell, une interface en ligne de commandes à destination de la GUI. Cela fut réalisé entre juillet et début août.

Le fonctionnement de notre équipe était basé sur des discussions d'un rythme de plusieurs par semaines au début, lors de la définition de la structure, pour passer à environ une toute les une ou deux semaines, principalement lorsqu'il s'agissait de fusionner nos travaux. Un système de suivi de version fut utilisé et un document de suivi des bugs fut instauré en juillet.

Validation de MOST et résultats obtenus

Avec le développement de MOST a suivi sa validation vis à vis de l'ancienne version, MOST sur OPNET. Des tests

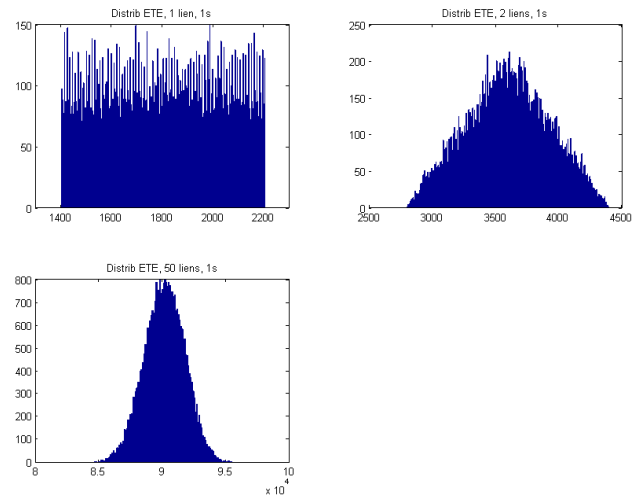


Figure 4: Distributions des temps de propagations des timecodes pour 1, 2 et 50 lien(s) à parcourir (Nombre total de timecodes envoyés inconnus, durée de la simulation : 1 seconde)

unitaires ont été menés pour vérifier le comportement ainsi que des scénarios sur des architectures réelles (comme sur celle de *Meteo Third Generation*, un satellite utilisant un réseau SpaceWire complexe). Ces validations, bien que non encore terminées, ont permis de corriger un certain nombre de comportements incorrects sur MOST mais également sur l'ancienne version. L'utilisation de MOST a également permis de vérifier des résultats de comportement attendus du standard SpaceWire. Par exemple en figure 4 sont présentés les répartitions des temps de propagation des timecodes lorsqu'ils parcourent 1, 2 et 50 liens. Pour le passage d'un lien, le temps de propagation d'un timecode est une variable aléatoire uniforme puisque le timecode doit attendre la fin de la transmission du caractère actuel. Ainsi, en parcourant deux liens on observe bien une distribution triangulaire, et pour 50 liens, la distribution se rapproche d'une distribution normale, par application du théorème central limite.

Apport et conclusion

Ce stage au sein de l'équipe de recherche et développement en avionique et plateforme m'a permis de développer mes capacités en modélisation des systèmes, standards et procédures complexes. J'ai par ailleurs pu me familiariser avec le domaine des *Data Handling Systems* et les problématiques temps-réel. J'ai de plus pu mener des analyses approfondies sur le standard SpaceWire, appelé à se développer dans les années à suivre. J'ai enfin renforcé mes capacités en gestion de projet informatique complexe en équipe.