Réseaux maillés dynamiques pour applications temps réel à criticités multiples : problématiques et analyse

Florian Greff, Ye-Qiong Song, Arnaud Samama, Laurent Ciarletta

Réseaux maillés dynamiques pour applications temps réel à criticités multiples : problématiques et analyse

Florian Greff
LORIA - Université de Lorraine
Thales Research & Technology
florian.greff@thalesgroup.com

Ye-Qiong Song
LORIA
Université de Lorraine
ye-qiong.song@loria.fr

Arnaud Samama
Thales Research & Technology
arnaud.samama@thalesgroup.com

Laurent Ciarletta
LORIA
Université de Lorraine
laurent.ciarletta@loria.fr

***Abstract*—Dans le cadre d'une thèse CIFRE chez Thales Research & Technology et au LORIA, débutée en 2015, nous étudions le chargement dynamique et la dégradation contrôlée d'un ensemble applicatif distribué, à criticités multiples, sur un réseau maillé. Il y a aujourd'hui une demande croissante en termes de systèmes temps réel distribués capables de supporter des ensembles applicatifs à criticités multiples. Dans les contextes de l'avionique ou du spatial par exemple, les contraintes de sûreté et de temps réel se superposent à celles liées aux coûts, à la consommation énergétique ou encore à l'espace disponible au sein du véhicule. Nous envisageons une nouvelle approche basée sur un réseau maillé de calculateurs, qui soit à la fois résilient et auto-reconfigurable de façon dynamique. Nos travaux s'intéressent au comportement de ce réseau ainsi qu'à sa modélisation, afin de démontrer sa viabilité pour supporter des ensembles applicatifs embarqués temps réel à criticité mixte. Dans ce document, nous proposons un aperçu des technologies d'interconnexion existantes et les confrontons à nos besoins, afin d'analyser les problématiques apportées par notre projet.**

I. Introduction

Le monde des systèmes embarqués (avionique, vétronique, drones, ...) connaît une réelle évolution des besoins en termes de puissances de calcul et de communications entre applications. Cela est induit par l'évolution des qualités des capteurs, dont le traitement des données s'alourdit, ainsi que l'apparition de nouvelles applications telles que des applications multimédia auparavant absentes de ce genre de systèmes. L'interaction entre les applications augmente, les fonctionnalités ont tendance à se répartir sur plusieurs composants de calcul.

D'un autre côté, ce monde est soumis à de très fortes contraintes telles que le caractère temps réel dur de certaines applications, leur caractère critique (mise en danger de la mission et/ou de vies humaines en cas de dysfonctionnement) ainsi que les contraintes matérielles telles que la consommation énergétique, la place disponible au sein du véhicule ou encore la présence d'un environnement disruptif.

Nous envisageons une nouvelle approche pour la mise en place d'ensembles applicatifs soumis à ces contraintes sur un groupe de composants matériels, basée sur la mise en réseau maillé de ces composants (Fig. 1). A l'aide de la virtualisation des calculateurs et/ou du réseau les reliant, cette plateforme doit permettre :

- une tolérance aux fautes par la réallocation des fonctions en cas de panne de leur calculateur d'origine et/ou la dégradation contrôlée des fonctions moins critiques si le total des ressources disponibles n'est plus suffisant ;
- une optimisation à chaque instant de l'utilisation des ressources par la négociation de contrats de Qualité de Service (QdS) et la dégradation contrôlée abordée ci-dessus ;
- une robustesse des communications inter-applications grâce à la topologie maillée et un protocole de routage adaptatif.

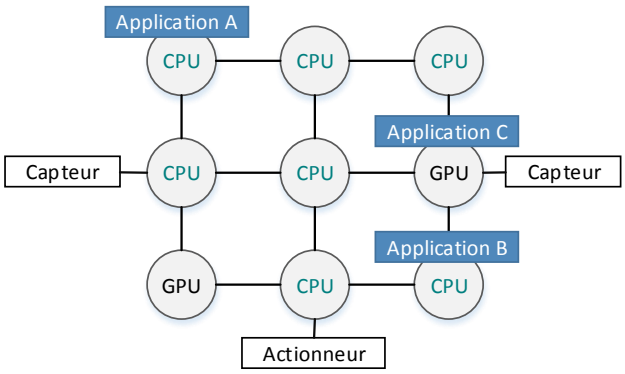

Fig. 1. Illustration de la plateforme visée

Ces fonctionnalités doivent de plus respecter les contrats de QdS en présence à chaque instant, notamment en ce qui concerne les propriétés temps réel des communications critiques. Ainsi, les travaux de recherche consistent, d'une part, à identifier les propriétés réseau essentielles au bon fonctionnement du système distribué, par l'étude de solutions d'interconnexion existantes ; d'autre part, à définir un formalisme permettant au concepteur d'application d'utiliser la plateforme de façon transparente.

La suite de ce document est divisée en deux parties : dans un premier temps, nous proposons un aperçu des solutions d'interconnexion existantes. Puis, nous les mettons en comparaison avec nos objectifs initiaux afin d'expliciter deux enjeux scientifiques majeurs inhérents à notre projet.

II. Solutions d'interconnexion pour l'embarqué temps réel

A. Généralités

Le monde de l'Internet est largement gouverné par la pile protocolaire TCP/IP, avec une liaison câblée s'opérant grâce à Ethernet. Cette solution d'interconnexion est satisfaisante dans un environnement tel qu'Internet où le réseau dispose de bonnes propriétés, sans contrainte temps réel dur, critique (mise en danger d'êtres humains si des données parviennent à destination trop tardivement).

Cependant, son utilisation dans l'embarqué pose deux limitations principales :

1) Il n'existe aucun mécanisme permettant de garantir le respect des contraintes temps réel dures (respect impératif des délais) ;
2) Il est difficile de rendre les implémentations logicielles déterministes. Quant aux implémentations matérielles, leur complexité rend leur certification impossible.

Partant de ce constat, des solutions d'interconnexion alternatives ont été conçues pour répondre aux besoins spécifiques de l'embarqué temps réel. Elles se divisent globalement en deux catégories, par bus ou commutées (Fig. 2).

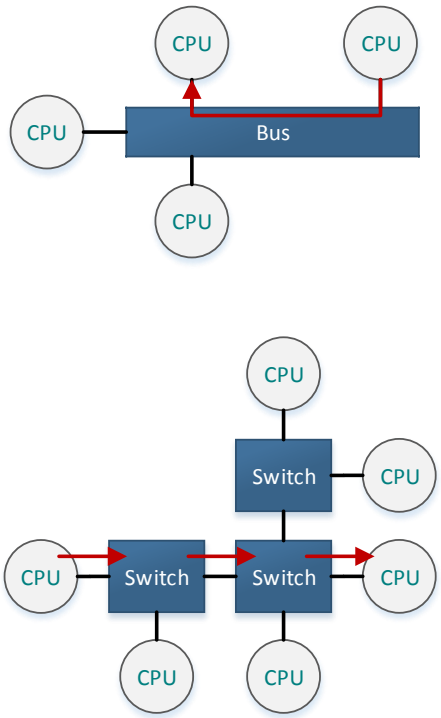

Fig. 2. Bus versus réseau commuté

B. Interconnexions par bus

L'interconnexion se fait par une voie unique bien que des ponts et/ou architectures hiérarchiques puissent être mis en place pour les systèmes les plus larges. On peut citer pour exemples PCI [1], CAN [2], ARINC 429 [3]. Ces technologies ont été éprouvées avec succès mais la présence d'un point unique de défaillance oblige une redondance physique, ce qui induit des coûts élevés. De plus, la centralisation du canal physique augmente le risque de collisions entre flux. Ainsi, un système d'arbitrage central doit être présent afin d'ordonner les flux entre eux, d'où l'apparition d'un délai de communication croissant avec le nombre de composants interconnectés. Pour cette raison notamment, le passage à l'échelle est problématique.

C. Interconnexions commutées

Le trafic réseau est réparti sur plusieurs liens physiques reliant les nœuds entre eux. De l'expéditeur au destinataire, les paquets de données empruntent un chemin défini à l'émission ou au fur et à mesure de leur progression dans le réseau. Ethernet est un exemple de protocole permettant l'interconnexion commutée au sein d'un réseau local. Pour ce qui est des réseaux temps réel, plusieurs grands standards ont émergé ces dernières années.

AFDX est un réseau avionique redondant implémentant le standard ARINC 664 et s'appuyant sur la technologie d'Ethernet commuté *full-duplex*. Il est notamment utilisé à bord de l'Airbus A380 [4]. Son déterminisme est assuré par une réservation de bande passante pour chaque flux, grâce à un système de canaux virtuels. Cela permet une ségrégation des flux traversant le même canal physique et garantit une maîtrise de l'utilisation qui est fait du réseau par les composants interconnectés. La redondance est apportée par la duplication des éléments réseau (canaux, commutateurs...). AFDX est un protocole éprouvé avec succès mais l'obtention du déterminisme impose la définition de liens virtuels statiques, ce qui le rend trop rigide par rapport à nos besoins de reconfiguration dynamique. De plus, la fiabilité du réseau repose sur les commutateurs dédiés, ce qui va à l'encontre de la philosophie distribuée que nous essayons de mettre en œuvre. Enfin, la duplication de chaque élément rend le passage à l'échelle complexe.

PCI Express [5] est un standard dérivé de la norme PCI mais fonctionnant de façon commutée. Dans le monde de l'embarqué, il hérite du succès de cette dernière tout en proposant un gain de performance et de disponibilité ainsi qu'une détection d'erreur plus élaborée. Néanmoins, dans notre cas, il pose des problèmes de topologie puisque les connexions sont très hiérarchisées (les composants racine émettent vers les enfants). **Advanced Switching Interconnect** [6] est un protocole d'encapsulation se présentant comme une solution à cette limitation. Il permet de créer des réseaux de n'importe quelle topologie, notamment sur PCI Express.

RapidIO [7] [8] est un standard initialement pensé pour l'embarqué. Il est utilisé dans la plupart des stations 3G/4G (système interne), dans l'imagerie médicale et le calcul hautes performances. Ses objectifs sont un débit élevé, une réduction maximale de la latence ainsi que sa garantie, le tout en disposant de communications fiables. La latence faible est atteinte grâce à l'utilisation de symboles de contrôle pouvant être embarqués dans les paquets de données. Les symboles de contrôle sont de petits en-têtes protocolaires utilisés notamment pour le contrôle de flux. Grâce à la possibilité de

les embarquer dans les paquets de données, c'est-à-dire les transmettre immédiatement pendant le transfert d'un paquet (sans attendre), on peut atteindre des latence très faibles pour la transmission d'informations de contrôle entre deux nœuds. Cela permet par exemple de garantir une retransmission extrêmement rapide en cas d'erreur, et donc une latence bout-en-bout améliorée pour les flux temps réel.
RapidIO permet également d'assigner des priorités et/ou canaux virtuels aux flux. En revanche, les politiques de traitement de ces priorités et d'ordonnancement en sortie des commutateurs doivent être définies par l'utilisateur. Un contrôle de flux bout-en-bout peut être effectué selon divers moyens (basé sur des crédits, XON/XOFF, etc.), ce qui paraît très intéressant pour l'implémentation d'un contrôle dynamique de la QdS tel que nous le concevons dans notre projet. Enfin, RapidIO propose des formats de registres pour la configuration des commutateurs et la lecture de leurs attributs, un algorithme de découverte du réseau et un mécanisme de détection des fautes très rapide.

SpaceWire [9] est, comme son nom l'indique, une technologie dédiée au spatial, utilisée ou envisagée dans de grandes missions spatiales : ExoMars, Swift, Rosetta, Lunar Orbiter... Cette orientation se traduit notamment par l'utilisation du *wormhole routing*, c'est-à-dire qu'un paquet peut s'étaler sur plusieurs tampons le long du chemin. En effet, les contraintes environnementales du spatial induisent des tampons très petits, voire qui ne permettent même pas la mise en tampon d'un paquet entier. Cependant, cette technique peut favoriser l'apparition d'interblocages puisque les paquets occupent plusieurs tampons simultanément même si leur progression dans le réseau est bloquée. Le routage est effectué soit de façon classique, grâce à l'adresse du destinataire ainsi que des tables de routage, soit par la spécification du chemin complet au niveau de l'émetteur. Cette dernière technique à tendance à centraliser le routage mais améliore ainsi son contrôle.
Globalement, on constate à la lecture du standard une philosophie générale proche de RapidIO, notamment en ce qui concerne la fiabilisation des communications ainsi que le contrôle de flux point-à-point. Néanmoins, certaines fonctionnalités ne sont pas spécifiées : configuration des commutateurs, gestion des priorités, multidiffusion, notification de paquets perdus au niveau de la couche applicative de l'émetteur. Enfin, SpaceWire spécifie un protocole de synchronisation des horloges, dont nous avons besoin pour certaines applications qui nous intéressent, radar par exemple.

InfiniBand [10] est utilisé et reconnu dans le monde du calcul hautes performances où il permet des échanges très rapides. Néanmoins, la complexité de sa mise en œuvre, l'opacité de sa politique d'arbitrage ainsi que l'incompatibilité des câbles utilisés avec une utilisation embarquée posent de trop grandes limitations par rapport à notre projet.

En synthèse de l'étude de ces standards, on retient qu'ils sont fédérés par trois objectifs forts :

1) Maîtrise de la **latence** bout-en-bout ;
2) **Fiabilité** bout-en-bout des transmissions ;
3) Tolérance de la **criticité multiple** (co-existance de flux critiques et déterministes et de flux temps réel souple et/ou moins déterministes), par la maîtrise de l'utilisation faite du réseau par chaque flux.

A une échelle plus petite, les *Networks-on-Chip* [11] [12] constituent également une référence intéressante puisque les contraintes d'utilisation du réseau maillé, notamment en termes de gestion de la congestion du réseau, sont analogues.

III. Enjeux scientifiques

L'objectif de nos travaux est de permettre à un système applicatif entièrement dynamique – par le biais des réallocations, de la virtualisation des composants de calcul et de la dégradation contrôlée – de communiquer sur un réseau maillé, tout en conservant des propriétés temps réel fortes.

Bien que les technologies présentées ci-dessus garantissent des communications fiables et optimisées en termes de latence, nous identifions deux défis majeurs apportés par notre projet.

Le premier découle du caractère dynamique de la localisation des applications sur le réseau. On s'intéresse plus particulièrement aux transitions entre les états du système applicatif. Par exemple, on peut considérer :

- que l'on peut estimer de façon fiable les propriétés réseau (bandes passantes, latences) puis allouer les ressources de sorte à garantir à une application temps réel critique A la satisfaction de ses contraintes temps réel ;
- que l'on peut opérer une dégradation contrôlée sur cette application en cas d'apparition d'une nouvelle interférence (congestion) sur un lien qu'elle utilise, suite à la réallocation d'une application B (Fig. 3), de sorte à préserver la satisfaction de ses contraintes temps réel, par exemple en lui demandant de diminuer la fréquence de ses communications.

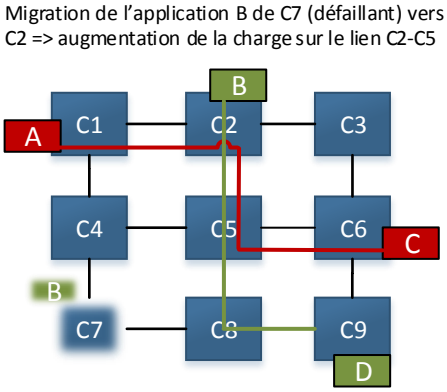

Fig. 3. Illustration d'une phase de transition

Néanmoins, on constate la présence d'une **phase de transition** le temps que la renégociation des ressources réseau allouées à A s'opère, durant laquelle l'augmentation de la charge sur le lien due à la présence de B (Fig. 3) peut conduire à une violation de la contrainte temps réel dure de A, ce qui est inacceptable. Nous travaillons donc à garantir la consistance du comportement du réseau lors des phases de transition, de sorte

que les contrats de QdS associés à chaque application soient respectés en chaque instant, malgré le caractère fortement dynamique de notre système. Cela doit se concrétiser par la définition de propriétés réseau (algorithme d'ordonnancement en sortie des nœuds, fonctionnement des protocoles, routage adaptatif).

Le deuxième défi est relatif à la formalisation de l'utilisation du réseau et de l'expression des besoins (Qualité de Service), afin de pouvoir mapper les propriétés introduites ci-dessus avec des mécanismes d'intergiciels et/ou des outils d'aide au développement.

En effet, la notion de Qualité de Service est présente à de nombreux niveaux de notre système :

- le développeur doit pouvoir exprimer ses besoins applicatifs, en termes de puissance de calcul et d'utilisation du réseau ;
- les contrats de QdS négociés pour chaque application doivent être modélisés ;
- si un intergiciel (par exemple, DDS [13]) est utilisé pour faciliter la liaison entre cette partie purement logique et le réseau effectif, il peut également permettre la spécification de contraintes de QdS ;
- les technologies d'interconnexion introduites en II-C peuvent elles aussi posséder des mécanismes de gestion de la QdS (contrôle de flux par exemple).

Il est nécessaire de traduire les contrats de QdS définis au plus haut niveau en propriétés concrètes (ordonnancement, champs protocolaires, routage, algorithme de contrôle d'admission), mises en évidence lors de la première partie (cf. plus haut). Or, bien que les technologies d'intergiciel et d'interconnexion proposent parfois un support de la gestion de la QdS temps réel (par exemple, une option dans le protocole), son implémentation reste à la charge du concepteur. Ainsi, nous travaillons à définir un formalisme faisant le lien entre les besoins de l'application, les contrats logiques négociés, l'intergiciel permettant l'accès au réseau et les propriétés réseau concrètes. Une contribution pourra par exemple être de définir un mappage entre les contraintes de QdS supportées par un intergiciel et la technologie d'interconnexion sous-jacente, l'ordonnancement des paquets réseau et/ou leur priorité dans l'entête protocolaire, etc.

IV. Conclusion

La mise en réseau maillé de calculateurs standards apparaît comme une solution élégante au problème de tolérance aux fautes adressé par les systèmes embarqués temps réel critiques, grâce à la mutualisation de la redondance physique ainsi que la pluralité naturelle des chemins de communication. Dans ce document, nous avons présenté plusieurs technologies permettant l'interconnexion des nœuds selon des mécanismes optimisés pour des systèmes temps réel critiques. Nous avons également présenté les défis qu'elles posent, compte tenu du caractère dynamique de la plateforme visée. L'étude des phases de transition entre états du système doit permettre de dégager des propriétés réseau concrètes, essentielles à la tolérance de ce dynamisme. Enfin, la définition d'un formalisme permettant de lier les contraintes de QdS d'un niveau de logique à l'autre est un objectif très motivant.

References

[1] PCI Special Interest Group, "Conventional PCI 3.0 Specification," 2004. [Online]. Available: https://www.pcisig.com/specifications/conventional/s

[2] "Can in Automation Other Specifications," Nov. 2002. [Online]. Available: http://www.can-cia.org

[3] AIM GmbH, "ARINC429 Specification Tutorial," Nov. 2010. [Online]. Available: http://www.aim-online.com/pdf/OVIEW429.PDF

[4] F. Brajou and P. Ricco, "The Airbus A380 - an AFDX-based flight test computer concept," in *AUTOTESTCON 2004. Proceedings*, Sep. 2004, pp. 460–463.

[5] PCI Special Interest Group, "PCI Express Base 3.1 Specification," Oct. 2014. [Online]. Available: https://www.pcisig.com/specifications/pciexpress

[6] S. Zirin and J. Bennett, "Advanced Switching Overview," PCI-SIG Developers Conference, 2004.

[7] RapidIO Trade Association, "RapidIO Specification 3.1," Oct. 2014. [Online]. Available: http://www.rapidio.org

[8] S. Fuller, *RapidIO: The Embedded System Interconnect*. John Wiley & Sons, Jan. 2005.

[9] S. Parkes, "SpaceWire User's Guide," 2012. [Online]. Available: https://www.star-dundee.com/knowledge-base/spacewire-users-guide

[10] Infiniband Trade Association, "Infiniband Architecture Specification, Release 1.3," Mar. 2015. [Online]. Available: http://www.infinibandta.org

[11] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[12] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-chip," *ACM Comput. Surv.*, vol. 38, no. 1, Jun. 2006. [Online]. Available: http://doi.acm.org/10.1145/1132952.1132953

[13] Object Management Group, "Data Distribution Service (DDS), Version 1.4," Apr. 2014. [Online]. Available: http://www.omg.org/spec/DDS/