

SpaceWireのリアルタイム性保証手法の検討とソフトウェアプラットフォームの開発(P5-062)

高田光隆, 陳暘, 高田広章(名古屋大学), 湯浅孝行, 高橋忠幸(JAXA), 能町正治(大阪大学)

SpaceWireは宇宙機向けネットワーク規格として国内外の科学衛星への採用が始まっている。SpaceWireをより多くの宇宙機や宇宙分野以外へも適用するために、SpaceWire上でのデータ転送にかかる時間を保証するためのリアルタイム性保証手法と、それをサポートするソフトウェアプラットフォームの開発が必要であると考えられる。

SpaceWireリアルタイム性保証手法検討会

名古屋大学, JAXA, 宇宙関連企業に呼びかけて、2011年6月より検討会を発足

<2011年度の主な活動>

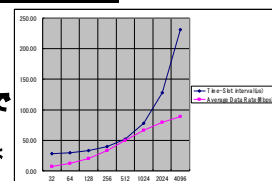
- SpaceWireのリアルタイム性保証に関する要件収集
- SpaceWireを採用したプロジェクトで得られた知見や課題の整理
- 成果: 「SpaceWireリアルタイム性保証手法ガイドライン」の策定

<2012年度の主な活動>

- SpaceWire-Dをベースとする拡張・追加した提案プロトコルの検討、策定
- 提案プロトコルをサポートするソフトウェアプラットフォームの開発

SpaceWire-Dの懸念点

- 実効転送レートが低い
例: リンクレート: 200Mbps
パケットサイズ: 256バイト
実効転送レート: 30Mbps (15%)
- RMAPターゲットがH/Wであることを前提に設計
ターゲットがソフトウェア実装の場合はさらにタイムスロット間隔が必要



RMAPのパケットサイズと実効転送レート

リアルタイム性保証手法による提案プロトコルの策定

- SpaceWireのリアルタイム性保証の考え方
 - ネットワークをタイムスロットと呼ばれる単位で時分割する
 - 各タイムスロットで1つのRMAPトランザクションのスケジューリングを予め決定する
 - スケジューリングの決定にはパケットの経路とそこで流れるパケットの最大滞留時間から算出
- SpaceWire-Dを拡張・追加した提案プロトコル
 - モードの概念の導入
 - サブネットの概念の導入 (図1)
 - 柔軟なタイムスロットの導入 (図2)
 - 分割RMAPトランザクションの導入
 - 任意の上位レイヤをサポート

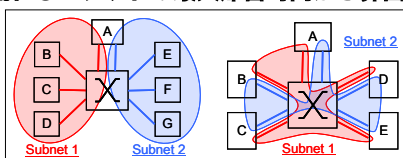


図1 サブネットの例

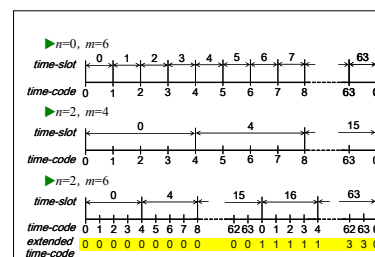


図2 タイムスロットの例

ソフトウェアプラットフォームの開発

SpaceWire ミドルウェアの開発

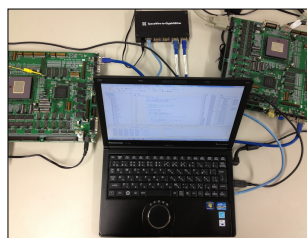
- 指定したタイムスロットにトランザクションを決定し送信するSpaceWire-D仕様をベースに拡張・追加した提案プロトコルを策定、開発
- RMAPトランザクション、SpaceWireパケットに導入可能
- RMAPパケットのメモリアクセス保護機能 (HRP2カーネル利用時)
- アプリケーションは時間管理を気にせずトランザクションを発行できる

高信頼性RTOS (TOPPERS/HRP2カーネル)の開発

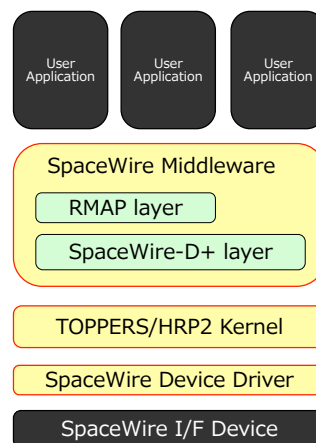
- JAXAと共同開発した高信頼RTOS (HRPカーネル) の次バージョン
- MMUだけでなくMPUを搭載したプロセッサもサポート
- プログラムのコードやメモリ配置をコンフィギュレータにより決定
- メモリ保護情報を静的に生成

SpaceWireデバイスドライバAPIの策定

- SpaceWireミドルウェア向けデバイスドライバAPIをボードメーカーと策定



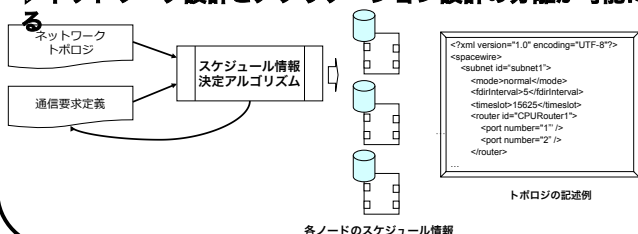
開発中のSpaceWire搭載ボード



ソフトウェアプラットフォーム階層図

スケジュール情報決定ツールの開発

- SpaceWireのネットワークトポロジと通信要求より、各ノードのタイムスロットへのRMAPトランザクションの割付を決定する支援ツール (外部ツール)
- 複雑なネットワーク設定のスケジュール情報を機械的に行う
- ネットワーク設計とアプリケーション設計の分離が可能になる



スケジュール情報決定アルゴリズム

焼きなまし法(Simulated Annealing)による最適解の探索

- 評価関数

$$OF(S) = (N_o + 1) \sum_{i \in S} U_i^S + N_o + \sum_{i \notin S} U_i^S$$

- U_i^S : 解Sでslot i の使用率
- ϕ : 全スロット群
- N_o : 全スロット数
- ϕ_s : 使用中スロット群
- N_s : 使用中スロット数
- ϕ_o : オーバーフロースロット群

OF(S)の結果は小さい方が良い解、逆に大きい方が悪い解と判断する

