

ROS の Publish/Subscribe 通信のハードウェア実装による高速化の検討

菅田 悠平[†] 山科 和史^{††} 大川 猛^{††} 大津 金光^{††} 横田 隆史^{††}

[†]宇都宮大学工学部情報工学科 ^{††}宇都宮大学大学院工学研究科情報システム科学専攻

1 はじめに

近年、自律型ロボットの開発が盛んに行われ、災害や日常生活の支援などでの活躍が期待されている。自律型ロボットは、自律制御や画像認識といった多くの処理をバッテリー駆動で行う必要がある。バッテリー駆動という電力面での制約がある中で、高性能に処理を行うには、マイクロプロセッサや GPU よりも電力当たりの性能が高い FPGA が適している。しかし、FPGA は開発コストが高いため、ロボット開発に導入することは容易ではない。この課題に対して、我々は ROS 準拠 FPGA コンポーネントを提案している [1]。ROS (Robot Operating System) とは、コンポーネント指向開発を支援している、ロボット開発のソフトウェアプラットフォームである。ROS に準拠して、FPGA をコンポーネント化することで、FPGA を容易にロボットシステムに導入することが可能となる。FPGA を導入することによって、電力当たりの性能が高いロボットシステムを実現できる。しかし現在のコンポーネントは、通信における遅延時間が大きいことが課題となっている。

本研究では、ROS の通信の高速化を目的として、従来ソフトウェア上で行っていた ROS の Publish/Subscribe 通信のハードウェア化を提案する。本稿では、ハードウェア化の検討及び実装方法について述べる。

2 ROS 準拠 FPGA コンポーネント

ROS 準拠 FPGA コンポーネントとは ROS の Publish/Subscribe 通信に準拠し、FPGA をコンポーネント化することである。コンポーネント化することにより、FPGA を ROS のシステムにおいてひとつのノードとして扱うことができる。そのためロボットシステムにおいて FPGA の回路の追加・変更が容易に行える。

ROS の Publish/Subscribe 通信とは ROS が提供するノード間の通信モデルの一つである。ROS におけるノードとは機能毎に実装されたプロセスのことである。Publish/Subscribe 通信とは、Publisher (出版者) と Subscriber (購読者) による非同期な通信方法である。Publisher は Topic というメッセージの集合を配信 (送信) し、Subscriber は必要な Topic のみ購読 (受信) する。

この通信方法は相手の情報を必要としないため、動的なネットワーク構成が可能となる。そのため、複数のノードで構成されるロボットシステムにおいて、ノードの参入、変更、脱退を容易に行うことができる。

ROS 準拠 FPGA コンポーネントの概要図を図 1 に示す。コンポーネントは、プログラマブル SoC (ARM プロセッサ + FPGA) を用いて実装を行う。ARM プロセッサ上の ROS ノード (interface FPGA) が FPGA 及び他の ROS ノードとの通信を行う。現在のコンポーネントでは、FPGA での処理時間に比べて、通信における遅延時間が大きいために、FPGA での処理能力を活かしきれていないことが課題である [1]。

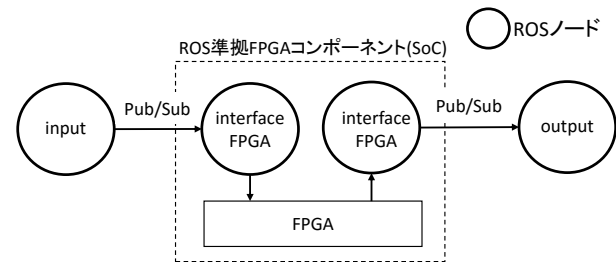


図 1: ROS 準拠 FPGA コンポーネント

3 Publish/Subscribe 通信のハードウェア化の検討

上記の課題を解決するために、コンポーネントの通信に使われている ROS の Publish/Subscribe 通信をハードウェア化することで通信の高速化を図る。

Publish/Subscribe 通信の詳細について述べる。Publisher, Subscriber, Master から成る、ROS の最小構成の通信手順を図 2 を用いて述べる。Master とは ROS のシステムを構築する際に必要不可欠なものであり、ノード情報の管理や、XMLRPC を元にした API の提供を行う。XMLRPC とは XML を HTTP で転送する、遠隔手続呼出のことである。

(1) Publisher ノードと Subscriber ノードは、起動すると自身のノード名や (配信又は購読する) トピックなどの自ノードの情報を Master に対して送信する。(2) Master は Subscriber が購読する Topic を配信している Publisher の情報を Subscriber に送信する。(3) Subscriber は Publisher に対して接続を要求する。このときの接続要求には接続方式の指定が含まれる。接続方式には TCP/IP ベースの TCPROS と UDP/IP ベースの UDPROS がある。(4) Publisher は接続要求に対して、自身のポート番号と URI アドレスを返す。(5) Subscriber は、その URI アドレスに対して、TCP

Study on hardware acceleration of Publish/Subscribe messaging of ROS

[†]Yuhei Sugata

^{††}Kazushi Yamashina, ^{††}Takeshi Ohkawa, ^{††}Kanemitsu Ootsu, ^{††}Takashi Yokota

Department of Information Science, Faculty of Engineering, Utsunomiya University ([†])

Department of Information Systems Science, Graduate School of Engineering, Utsunomiya University (^{††})

又は UDP 通信でコネクションを確立し, connection header を送る. (6)connection header を基に, Publisher ノードは, Publish/Subscribe 通信かサービス通信かを判断し, connection header を送り返す. (7) データの送信 (配信) をする. 以後, 同じコネクションを使用して (7) が繰り返される.

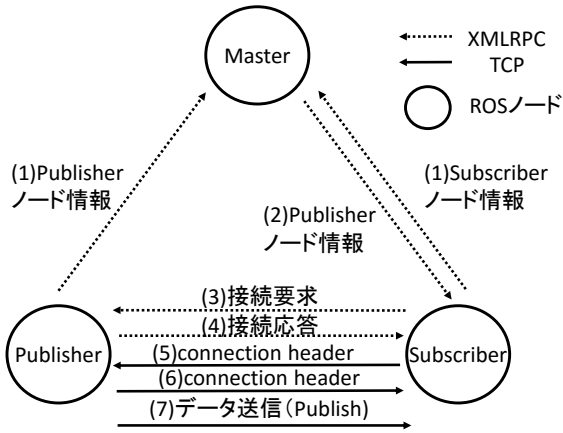


図 2: Publish/Subscribe 通信の手順

図 2 に示した ROS の最小構成における, TCPROS 接続の Publish/Subscribe 通信の packet 解析を行った. 解析には packet 解析ソフトウェアである Wireshark を用いた. ROS の接続方式では, TCPROS を使用するのが一般的である. 以後, 本稿では TCPROS を対象に検討を行う.

(1) ~ (4) の手順においては 229 個の packet を送出する. (5), (6) の手順では 7 個の packet を送出し, (7) では 2 個の packet が送出される. 手順 (1) ~ (6) はコネクションが確立するまでの 1 回のみであるのに対して, 手順 (7) はノードがデータの配信又は購読を止めるまで繰り返される. そのため, Publish/Subscribe 通信でボトルネックとなっているのは, 手順 (7) である. ゆえに, 手順 (1) ~ (4) で使われている XMLRPC の通信をハードウェア化しても, 通信の高速化は見込めない.

以上のことから, TCP によるデータ通信を行っている部分をハードウェア化することが通信の高速化に繋がると考えられる.

4 実装方法

提案する, Publish/Subscribe 通信ハードウェアの実装方法を検討した. データ通信部を実装するには, TCP/IP 通信をハードウェアで行う必要がある. 今回は, SiTCP[2] を用いて TCP/IP をハードウェア実装した. SiTCP とはハードウェア実装されたネットワークプロセッサであり, FPGA で TCP 通信を行うことが可能になる. アカデミック用途に限り無償公開され, 多くの FPGA に対応している.

実装した Publisher ノードのブロック図を図 3 に示

す. TCP/IP 通信を行うための SiTCP は GMII を介して PHY チップと接続する. SiTCP においては通信データの入出力には FIFO を用いる. 入力用 FIFO から読み出すことで packet を受信し, 出力用 FIFO に書き込むことで packet を送信する. Publisher データ生成ブロックでは図 2 で示した通信手順の (6)connection header の書き込みと (7) 送信データの書き込みを行う.

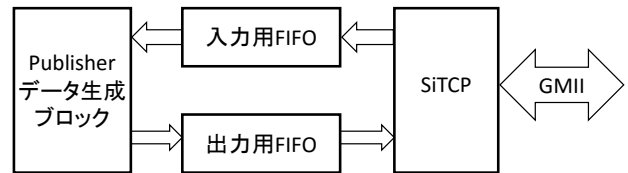


図 3: ハードウェアブロック図

動作確認のため, 図 3 の構成の Publisher ハードウェア回路を Xilinx 社の FPGA チップ Spartan-6 搭載の DIGILENT 社製の Atlys ボードに実装した. Subscriber と Master は PC 上で動作させた. PC 上の Subscriber ノードと FPGA 上に実装した Publisher ノードが Publish/Subscribe 通信を行うことが確認できた.

5 おわりに

本稿では ROS の Publish/Subscribe 通信のハードウェア化による通信の高速化の検討を行った. Publish/Subscribe 通信は, 登録処理を行う XMLRPC 通信部分と TCP/IP によるデータ通信部分からなる. Publish/Subscribe 通信を高速化するにはデータ通信部分をハードウェア化する必要があることを述べた. また, 検討を基に Publisher ノードのハードウェア化を行い, 動作確認を行った. 今後は Subscriber ノードをハードウェア実装し, 性能評価を行い, 従来の ROS 準拠 FPGA コンポーネントと通信の遅延時間の比較を行う予定である.

謝辞

本研究開発は, 総務省 SCOPE (受付番号 152103014) の委託を受けたものです.

参考文献

- [1] K.Yamashina, T.Ohkawa, K.Ootsu, T.Yokota: "Proposal of ROS-compliant FPGA Component for Low-Power Robotic Systems —case study on image processing application—," Proceedings of 2nd International Workshop on FPGAs for Software Programmers (FSP 2015), pp.62-67, 2015.
- [2] T.Uchida: "Hardware-Based TCP Processor for Gigabit Ethernet," IEEE Transactions on Nuclear Science, Vol.55, No.SIG 3, pp.1631-1637, 2008.