

ローカル 5G ネットワークにおけるロボット用ミドルウェア ROS 2 のリアルタイム性に関する初期評価

千家 雅之 (情報・生産技術部 システム技術グループ)

1. はじめに

ロボットの遠隔制御において無線通信を用いることは有線通信と比べてケーブル敷設作業が不要になる等利便性の面で大きなメリットを有するが、通信の即時応答性や安定性の面でデメリットがある。そのため、近年、超高速、多数同時接続、高信頼・低遅延を特徴とする 5G は大きな注目を集めており、産業分野での期待は大きい。ローカル 5G はモバイル向け通信規格である 5G に則った無線通信ネットワークであり、通信事業者以外の一般企業や自治体等によって構築・運用される。当研究所では、2020 年度にローカル 5G の無線局免許を取得し、2021 年度に海老名本部でローカル 5G の通信環境の運用を開始している。本稿では、ロボットの無線遠隔制御に焦点を当て、ロボット用ミドルウェアの中で大きなシェアを有している ROS (Robot Operating System)⁽¹⁾ の次世代バージョンである ROS 2 を対象に、当研究所のローカル 5G ネットワークを用いてリアルタイム性に関する測定と評価を行う。具体的には、ユーザ端末から物理的に近い場所にサーバー等のコンピュータを設置して応答性能を向上 (低遅延化) させるエッジコンピューティングを想定したシステム構成で、通信の応答性能を従来バージョンである ROS 1 と ROS 2 それぞれで測定し、リアルタイム性を評価する。

2. 要素技術の紹介

2.1 エッジコンピューティング

モバイル向け通信におけるエッジコンピューティングとは、基地局付近にエッジサーバーを設置し、機械学習や画像処理といった負荷が高い処理を基地局内で実施することで、遅延低減や帯域節約を図る手法のことである。エッジコンピューティングアーキテクチャの有用性は様々な研究で検証されている⁽²⁾。

2.2 ROS 1 と ROS 2 のアーキテクチャ

ROS 1 と ROS 2 はミドルウェアが担う範囲が異なっている。図 1 に示す通り、ROS 1 は独自のプロトコルである TCPROS/UDPROS 上にライブラリがあり、その上で ROS マスターが動作するという構成であるが、ROS 2 は通信の大部分を DDS (Data Distribution Service) が担い、共通のインターフェースを通して ROS ノードと DDS の橋渡しをするという構成である。ROS 2 は DDS の性能に依存する構成と言えるが、DDS が独立しているため別のベンダーの DDS に容易に変更することができる。

3. 評価方法

3.1 システム構成

評価システムのシステム構成は図 2 の通りである。ロボットに搭載されることを想定した PC #1 は LAN ケーブルを介してローカル 5G 端末に接続されている。エッジサーバーとして機能する PC #2 は LAN ケーブルでローカル 5G の上位ネットワークに接続されている。PC #1 および PC #2 の仕様を表 1 に、ソフトウェア構成を表 2 に示す。各 PC は ROS 1 インストール済み Linux と ROS 2 インストール済み Linux を PC 起動時に切り替えられる構成となっている。どちらの構成も OS の処理遅延を低減させるために low-latency カーネルを用い、さらに構成 B では DDS をリアルタイム性が高いとされる RTI 社製 DDS に変更している。

3.2 測定方法

ROS 1 および ROS 2 で用意されている ROS サービスリクエストを用いて、PC #1 と PC #2 間の応答性能を測定する。具体的には、クライアントである PC #1 から二つの整数を含む ROS サービスリクエストを送信し、サーバーである PC #2 で二つの整数を合計しレスポンスとして返すという処理において、ROS サービスの応答時間 (送信から受信までの所要時間) 及びその送信処理のジッター (送信タイ

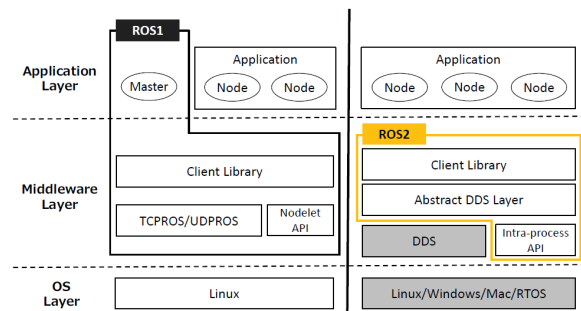


図1 ROS 1 と ROS 2 のアーキテクチャの比較
(文献⁽³⁾より引用)

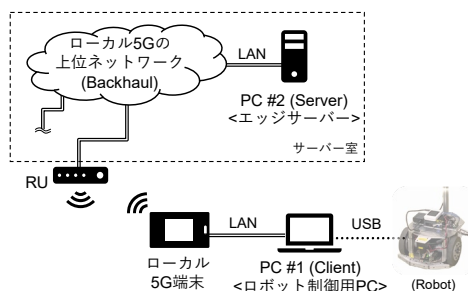


図2 評価システムの構成

ミングの揺らぎ) の測定を行う。送信処理のループは Rate クラスの sleep 関数を用い、周期は 500 ms と 100 ms の 2 パターンで、送信回数は 1000 回とする。

4. 結果及び考察

ROS サービスの応答時間をヒストグラムで図 3 に、その送信処理のジッターを箱ひげ図で図 4 に示す。

図 3 より、ROS サービスの応答時間は ROS 2 の方が短いことがわかる。ROS 1 では横軸 100 ms 前後に、ROS 2 では 50 ms 未満の区間にヒストグラムの棒が集中しているが、この ROS 1 と ROS 2 の違いは TCP ベースの独自プロトコルである TCPROS と UDP を用いる DDS との違いから生じているものと考えられる。また、図 3 (a) と (b) の応答時間の分布や最大値の違いは、通信環境の変動だけでなくクライアント側の送信処理での Rate クラスの sleep 関数の影響によるものと考えられる。この sleep 関数は既定の周期に沿うように次の周期までの残り時間分だけ待機する関数であるが、既定の周期を超えて残り時間が負になった場合は待機しない。つまり、サービスレスポンス受信後すぐにサービスリクエスト送信処理が行われるため、サーバー側で処理遅延が生じている可能性がある。

図 4 より、送信時のジッターは、周期 500 ms では ± 0.2 ms 程度で ROS 1 と ROS 2 の間に大きな差はないと考えられるが、周期 100 ms では ROS 1 の方が非常に大きい。これは ROS 1 では ROS サービスの応答時間が周期 100 ms を超える場合があるためである。なお、この測定は送信処理に入る前のタイムスタンプを利用しているため、厳密な比較のためにはパケットキャプチャ装置を用いる必要がある。

5. おわりに

評価システムを用いた測定より、ローカル 5G ネットワーク上で ROS の通信を行う場合、ROS 2 は ROS 1 よりも即時応答性が高いため、リアルタイム性の確保と短い周期での通信に利点があることがわかった。当研究所のローカル 5G ではエッジコンピューティングのアーキテクチャと ROS 2 を用いることで周期 100 ms 程度のリアルタイム通信が可能であることが分かった。

今後は、リアルタイム性の向上が期待できる ROS 2 の Executor クラス⁽⁴⁾を用いた検証やロボットに実装しての検証に取り組む予定である。

【参考文献】

1. Open Robotics: “ROS - Robot Operating System”, <https://www.ros.org/>
2. 山口弘純, 安本慶一, “エッジコンピューティング環境における知的分散データ処理の実現”, 電子情報通信学会論文誌 B, Vol.J101-B, No.5, pp.298-309 (2018)
3. Y. Maruyama, S. Kato and T. Azumi, “Exploring the Performance of ROS2”, 2016 International Conference on Embedded Software (EMSOFT 2016) (2016)
4. Y. Yuqing and T. Azumi, “Exploring Real-Time Executor on ROS 2”, In 2020 IEEE International Conference on

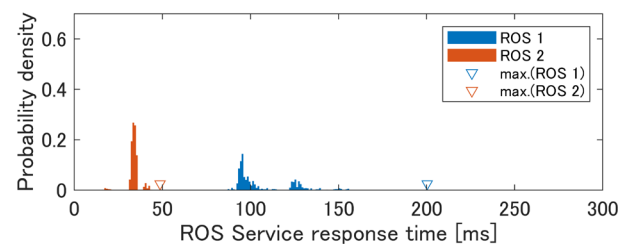
Embedded Software and Systems (ICESSE), 1–8 (2020)

表 1 PC の仕様

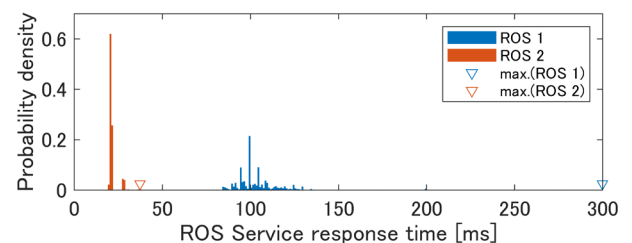
	PC #1 (Client)	PC #2 (Server)
Model	Intel NUC11 Pro	Shuttle SH370R6
CPU	Core i5-1145G7	Core i7-8700K
RAM	32GB	32GB
Disc	NVMe SSD 256GB	NVMe SSD 256GB

表 2 PC のソフトウェア構成

	構成 A	構成 B
OS	Ubuntu 18.04.6 LTS	Ubuntu 20.04.4 LTS
Kernel	5.4.0-104-lowlatency	5.13.0-35-lowlatency
ROS	ROS Melodic	ROS 2 Foxy
Protocol	TCPROS	RTI Connext DDS

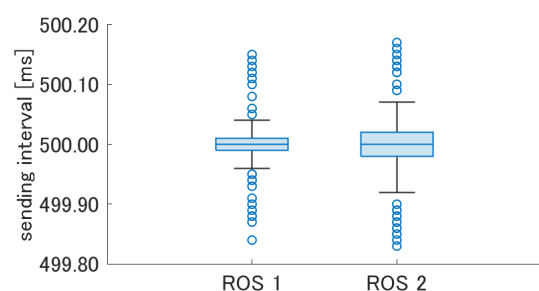


(a) 周期 500 ms の場合

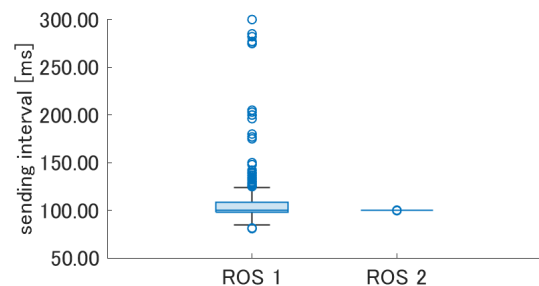


(b) 周期 100 ms の場合

図 3 ROS サービスの応答時間



(a) 周期 500 ms の場合



(b) 周期 100 ms の場合

図 4 サービスリクエスト送信時のジッター