

# 災害現場における複数台ロボット制御： データ共有とロボット指示を両立する ROS の一対一通信の拡張

小泉 亘平<sup>†</sup> 菅谷 みどり<sup>†</sup>  
 芝浦工業大学工学部情報工学科<sup>†</sup>

## 1. 研究の背景と目的

地震などの震災による倒壊建物内の要救助者探索において自律移動ロボットによる災害救助方式では、故障対策や広大な災害現場の効率的探索を実現するためロボットを複数台利用することが一般的となっている[1]. 複数台のロボット運用では、分散した無線通信が望ましい[2]. その無線通信手段としてブロードキャストやマルチキャストなどの手法が挙げられるが、信頼性や効率性の点で適しているとは言えない. そこでロボットミドルウェアの ROS (Robot Operating Systems) [3] が注目されている. ROS は信頼性の高い通信を保証すると同時に、Publisher/Subscriber モデルによる非同期通信により、効率的でかつ信頼性の高い通信を実現する. しかし、ROS では Topic と呼ばれる共有資源に対し、一律に登録ノードへの通信が発生する災害時には、特定のロボットへの命令指示を行うことが必須であることから、一律通信のみでは不十分である. 目的を達成するためには、ノードに対する通信と、一律通信を柔軟に切り替える手法が必要である.

本研究では ROS の一律にデータを送信するメリットを維持しつつ、同一フレームワークの中で特定のノードのみにデータを送付する一対一通信の仕組みを、従来の ROS の性能を損なうことなく実現した. 本論文では、ROS の基礎的な評価、および ROS の一対一通信の設計、実装、評価を行った.

## 2. 提案

### 2. 1 概要

提案方式では、ROS の一律の非同期通信である Publisher/Subscriber モデルの上に、一対一通信を可能とする Publisher と Subscriber の機能を必要に応じて持たせることで、サーバからは探索範囲の通信、クライアントからは要救助者発見と探索終了の通信を行う ROS 上のミドルウェアを提案する.

### 2. 2 Topic の配置方法

ROS の Publisher/Subscriber モデルでは、Topic を介して通信を行う非同期分散型のデータ共有が可能である. 本研究ではサーバとクライアントで一対一通信を実現するために Topic の分散を行った. その Topic の配置図を

図 1 に示す. Topic を分散することによりクライアント内で必要なメッセージか判断する必要がなくなり、特定のノードのみにデータ送付することが可能となった.

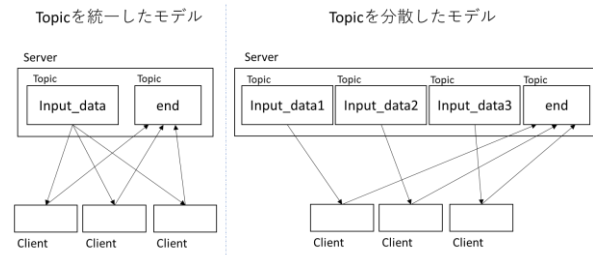


図 1 Topic 配置図

### 2. 3 機能設計

図 2 に設計のブロック図を示した. オペレータが探索範囲の入力を行うとサーバが領域分割を行い、それぞれのロボットの Topic に対して探索領域が Publish される. 監視している Topic が更新されると Subscribe しロボットが探索を始め、終わるとサーバに終了通知を送信する.

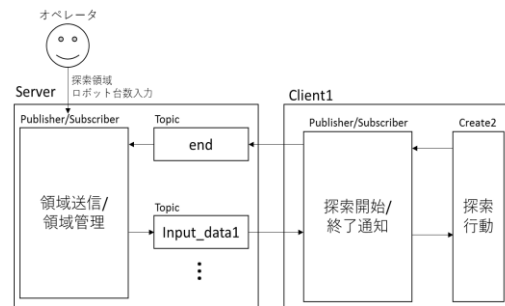


図 2 設計

## 3. 予備実験

### 3. 1 概要

基礎的な Publisher/Subscriber 通信の性能を測定することを目的に、比較対象としてブロードキャスト通信を用いて送信開始から受信終了までの時間を計測、Publisher/Subscriber モデルのネットワーク性能評価を行った. 実験環境は Intel Core i3 2. 1GHz, メモリ 4GB の PC で OS は Ubuntu14. 04 LTS ROS のバージョンは indigo を使用する.

### 3. 2 計測区間

ブロードキャスト方式はサーバが send() によりデータ送信を開始し、クライアントが recv() により受信終了までの区間、Publisher/Subscriber モデルは Publisher がデータを Publish し、Subscriber が

<sup>†</sup>Kohei Koizumi <sup>†</sup>Midori Sugaya <sup>†</sup>Department of Information Science and Engineering, College of Engineering, Shibaura Institute of Technology.

Subscribe()により Topic の更新を確認し, Callback()によりデータ受信終了するまでの区間とする.

### 3. 3 結果

ブロードキャスト方式と Publisher/Subscriber モデルについて, 800 回の計測時間の平均値・標準偏差値, 受信不可数を表 1 に, 送受信時間のヒストグラムを図 4 に示す.

表 1 データ共有時間の比較 (単位: ms)

	平均値	標準偏差値	受信不可数
ブロードキャスト方式	196. 8	3. 5	6
Publisher/Subscriber	465. 9	289. 0	0

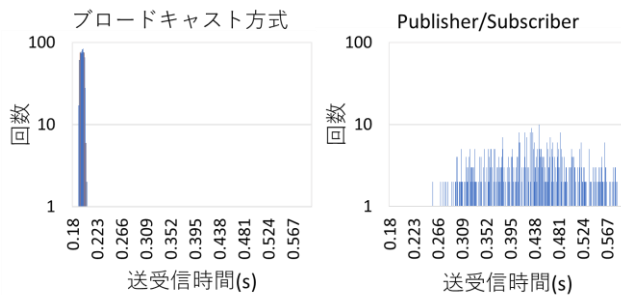


図 4 送受信時間のヒストグラム

### 3. 4 考察

実験結果より, 効率の点ではブロードキャスト方式のほうが優れているが, Publisher/Subscriber モデルは受信不可数がないため信頼性が高い. 平均値に差が出た理由としては, 計測区間のシーケンス数がブロードキャスト方式は1本なのに対し, Publisher/Subscriber モデルは3本であることが考えられる.

## 4. 評価

### 4. 1 概要

本研究の提案手法における台数増加による性能測定を目的とした実験を行う. サーバ側 PC は予備実験と同じものを使用し, クライアント側は Raspberry pi3 Model B, 探索ロボットとして iRobot 社の iRobot Create2 Programmable Robot を使用する.

### 4. 2 計測区間

実機実験における計測区間として図 5 に示す.

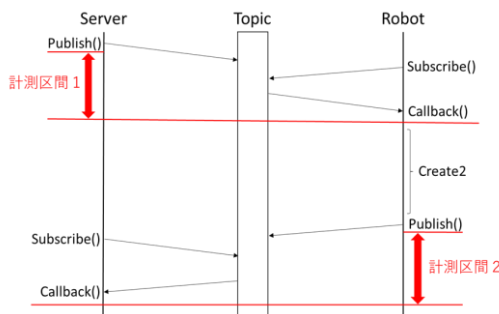


図 5 実機実験における計測区間  
オペレータが探索範囲を入力し, 領域分割が行われて

から測定を行う. 計測区間 1 はサーバがクライアントに対し探索領域を Publish し, Subscribe し終えるまでの区間, 計測区間 2 はクライアントが探索終了してからサーバが終了通知を Subscribe し終えるまでの区間とする.

### 4. 3 評価結果

台数増加によるサーバからクライアントへの Publish 時間とサーバのクライアントからの Subscribe 時間の平均値・標準偏差をグラフ化したものを図 6 に示す.

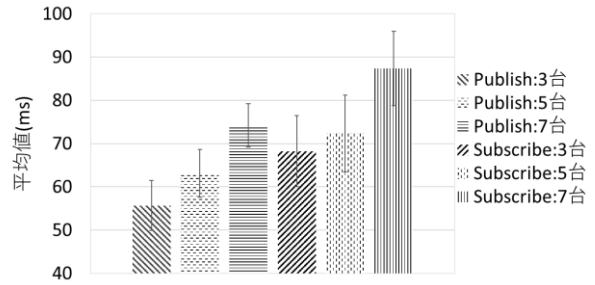


図 6 台数増加による平均値・標準偏差

Publish/Subscribe とともに台数が増加するに伴い, サーバからクライアントへの Publish 平均時間は 3 台から 5 台において 13.4%増加, 5 台から 7 台では 17.5%増加した. またサーバのクライアントからの Subscribe 平均時間は 3 台から 5 台において 6.0%増加, 5 台から 7 台において 20.8%増加した.

### 4. 4 考察

サーバが 1 台目のクライアントの終了通知を Subscribe し領域管理中の場合, 別のクライアントからの終了通知により Topic が更新されても監視していないため Subscribe することができない. 従って台数増加するに従い, クライアントからの終了通知が同時に Publish される可能性が高まり, 平均値の値が増加する.

## 5. まとめと今後の展望

考察における課題として, サーバのマルチスレッド化を行うことにより改善できると考える. また, 本研究の提案手法を用いることにより特定のクライアントのみに対し, メッセージを送信することが可能となった. したがって今後は, 実際の災害現場を想定した複数台ロボットにおけるフォーメーション制御に本研究の提案手法を用いることにより, より効率的な探索を行うことが可能になると考える.

## 参考文献

[1] R Murphy et al. "Mobility and Sensing Demands in USAR". 2000, IECON2000, Vol. 1, p. 138-142.  
 [2] 杉山久佳ほか. "ネットワーク化された群ロボットによる被災者発見システム". 情報処理学会論文誌. 2005, Vol. 46, No. 7. p. 1777-1778  
 [3] "About ROS". ROS.org. <http://www.ros.org/about-ros/> (参照 2018-01-11)