Performance Analysis of a Node and Module Discovery Protocol in Data Distribution Service

Ryosuke Tomishige[*1] Yuya Tarutani[*2] Tokumi Yokohira[*3]

Yuki Takano[*4] Koichi Imai[*5]

2025 年 12 月 5 日

[*1] Okayama University
[*2] The University of Osaka
[*3] Okayama University
[*4] TIER IV, Inc.
[*5] TIER IV, Inc.

0.1 イントロ

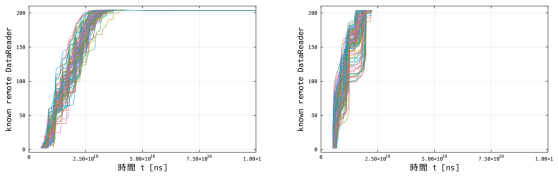
自動運転システムで採用されている ROS 2 は Data Distribution Service (DDS) と呼ばれる Pub/Sub 通信のミドルウェアを用いる。DDS ネットワークに参加するプロセスをノードと呼び、各ノードはトピックを出版/購読するためのモジュールを持つ。各モジュールは他のモジュールを Discovery Protocol (発見プロトコル) と呼ばれる仕組みを使用して自動的に発見し、通信を行なう。標準の発見プロトコルとして規定されている Simple Discovery Protocol (SDP) はノード発見とモジュール発見の 2 つの段階で構成される。しかし、SDP は小規模から中規模のネットワークを対象としており、大規模なネットワークで使用されることが想定されていない。

ROS2 でサポートされている DDS 実装のうち、eProsima’ s Fast DDS、Eclipse Cyclone DDS の 2 つについて性能を確認する。

0.2 Simpel Discovery Protocol

SDP のシーケンスの概略を図??に示す。各ノードは自身の情報を含む SPDP メッセージを定期的に送信し、ネットワーク上の他のノードに自身の存在を知らせる。SPDP メッセージを受信したノードは自身のモジュール情報を含む SEDP メッセージを送信する。

DDS のメッセージにはシーケンス番号が振られており、SEDP では信頼性確保のために、定期的に送信済み SEDP メッセージのシーケンス番号の一覧を HB メッセージで送信し、それを受信したノードは肯定確認応答または再送要求を ACKNACK メッセージで送信ノードに伝える。このとき、未知のノードからの SPDP 以外のメッセージは図??の二本目の矢印のように無視される。両ノードがお互いを発見した後に SEDP メッセージを受信することで他のノードが持つモジュールを発見する。



(a) Fast DDS (b) Cyclone DDS

図 1: Discovery Time

表 1: Discovery Time

Impl	recv buf size (MiB)	Discovery Time (s)
Fast	1	36
Fast	10	36
Cyclone	1	21
Cyclone	10	22

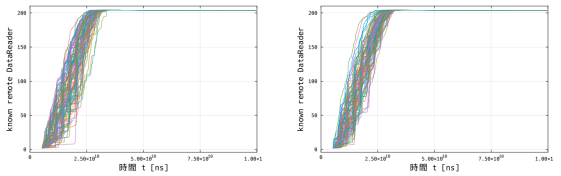
0.3 性能解析のための評価実験

すべてのノードが他ノードのもつ全モジュールの情報を受信するまでにかかる時間を Discovery Time と定義する。SDP のスケーラビリティを検証するために、車載ネットワークを模した多数のノードが存在するネットワークでノードを同時に立ち上げたときの、Discovery Time を計測する。2 台のホストを用意し、同一のトピックの出版/購読モジュールをそれぞれ 1 つ持つノードを Fast DDS, Cyclone DDS で実装し、各ホストで 102 ずつ立ち上げ実験を行なった。

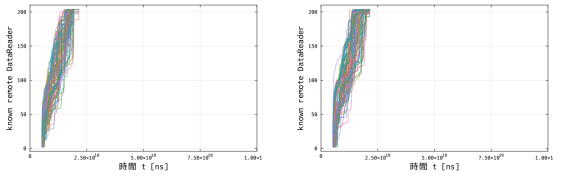
実験結果を図 4 に示す。Fast DDS は 45s、Cyclone DDS は 22s だった。Cyclone にはグラフに垂直なラインが見られるが、Fast には垂直なラインがみられない。

この差の原因として、UDP ソケットの受信バッファーが原因として考えられるため、各実装のソケット受信バッファーを変更して実験した。

どちらの実装でも 1MiB, 10MiB で変化無し。Cyclone はデフォルトが 1MiB なので明示的に指定しても変化無し。Fast はデフォルトではカーネルのデフォルト値を使用し、実験環境では 256KiB。1MiB に変更すると改善するがそれでも Cyclone に

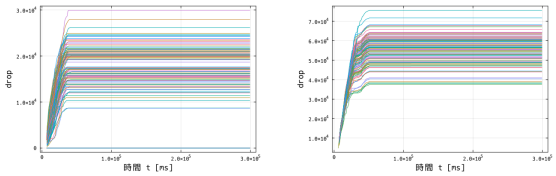


(a) Fast DDS 1MiB (b) Fast DDS 10MiB

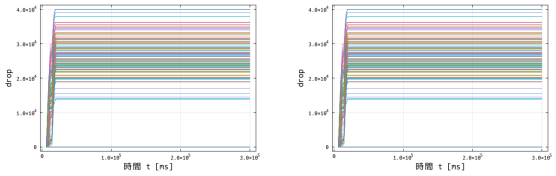


(c) Cyclone DDS 1MiB (d) Cyclone DDS 10MiB

図 2: Discovery Time

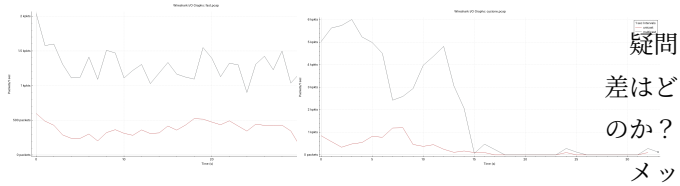


(a) Fast DDS Uni (b) Fast DDS Multi



(c) Cyclone DDS Uni (d) Cyclone DDS Multi

図 4: Discovery Time



(a) Fast DDS (b) Cyclone DDS

図 3: send message

比べて遅い。

1. なぜ Fast は Cyclone と比較してソケットバッファのサイズが同じでも Discovery が遅いのか？

あるノードに対して時間あたりに送信されるメッセージ数

シンプルに Fast DDS のメッセージ処理速度が遅い可能性が高い Fast DDS は意図的に 5ms のレイテンシを入れてるが、それよりも遥かに実測レイテンシが大きい。TODO: CPU 負荷の観点からチェック

Fast DDS: 100ms〜200ms

HB Multicast 1.873854

ACKNACK 1.906739

DATA(w) 2.124689

Cyclone DDS: ほぼ 0

HB Multicast 0.002968

ACKNACK 0.003874

DATA(w) 0.004259

疑問 2. Cyclone と Fast の discovery の進み方の差はどのような実装、パラメータの差によるものなのか？

メッセージの送信周期や送信方法、送信タイミングなどの差はあるが、影響はほとんど無さそう。すべて Fast DDS のメッセージ処理速度で説明できる。Cyclone DDS: DATA(w) の unicast, multicast の割合が 1:1 Fast DDS: DATA(w) の unicast, multicast の割合が 1:0

Fast でノード数を半分にするとどうなるのか？12s 程度で discovery 完了するようになる。グラフに垂直なラインがみられるようになった。

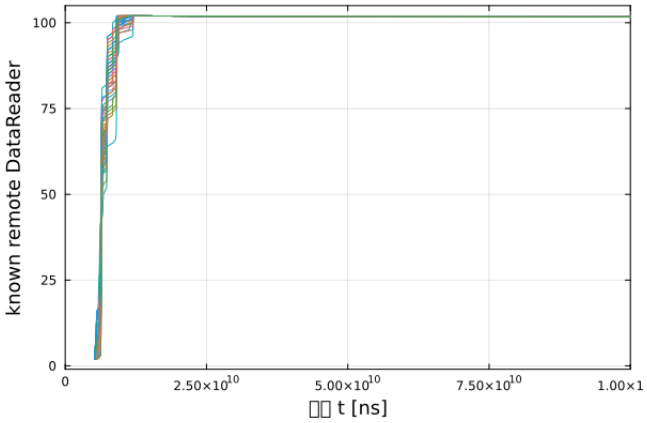


図 5: Fast DDS half node