

ОСОБЕННОСТИ ВЗАИМОДЕЙСТВИЯ С РОБОТОМ В РЕАЛЬНОСТИ И СИМУЛЯЦИИ

Содержание:

- DDS в ROS2
- Понятие daemon и discovery
- Конфигурация для работы с роботом
 - В симуляции
 - В реальности
- Trouble shooting, как проверить состояние сети/daemon

Основы передачи данных в ROS2: DDS, Discovery и сетевая настройка

1. Введение в архитектуру ROS2

ROS2 (Robot Operating System 2) — это современный фреймворк для разработки робототехнических систем. В отличие от ROS1, где был центральный мастер-сервер, ROS2 использует **децентрализованную архитектуру**, что делает систему более надёжной и гибкой.

Ключевое изменение: вместо единого roscore в ROS2 используется стандарт DDS для организации связи между компонентами.

2. DDS — фундамент коммуникации в ROS2

Что такое DDS?

DDS (Data Distribution Service) — это промышленный стандарт для обмена данными в реальном времени. В ROS2 DDS выполняет роль **транспортного слоя**, отвечая за всю сетевую коммуникацию.

Как работает DDS в ROS2?

- Каждый узел (нода) подключается к DDS-среде
- DDS автоматически обнаруживает другие узлы в сети
- Данные передаются напрямую между узлами (peer-to-peer)
- Поддерживаются различные качества обслуживания (QoS)

Популярные реализации DDS в ROS2: - Cyclone DDS (используется по умолчанию в последних версиях) - Fast DDS (ранее известный как FastRTPS) - Connex DDS

3. Discovery — механизм автоматического обнаружения

Что такое Discovery?

Discovery — это процесс **автоматического обнаружения** узлов в сети. Когда узел запускается, он “сообщает” о своём существовании, а другие узлы “откликаются” на это сообщение.

Как происходит Discovery?

1. **Приветственная фаза:** Узел рассылает multicast-сообщения о своём запуске
2. **Установление соединения:** Найденные узлы устанавливают прямые соединения
3. **Обмен данными:** Узлы обмениваются информацией о топиках, сервисах и действиях

Аналогия: Представьте вечеринку, где гости (узлы) приходят и знакомятся друг с другом без ведущего.

ROS_DOMAIN_ID — виртуальные сети

- Каждый ROS_DOMAIN_ID создаёт **изолированную сеть**
- Узлы с разными DOMAIN_ID не видят друг друга
- По умолчанию используется DOMAIN_ID=0

4. ROS2 Daemon — ускоритель работы

Назначение демона

ROS2 Daemon — это фоновый процесс, который **кэширует информацию** о топологии сети для ускорения работы командной строки.

Как работает демон?

- Запускается автоматически при первой ROS2-команде
- Участвует в discovery-процессе как обычный узел
- Сохраняет актуальную карту всех узлов и топиков
- Предоставляет быстрый доступ к информации через CLI

Пример использования:

```
ros2 node list    # Быстрый ответ благодаря демону
ros2 topic list   # Информация берётся из кэша демона
```

Перезапуск демона при проблемах:

```
ros2 daemon stop
ros2 daemon start
```

5. Настройка Cyclone DDS

Выбор сетевого интерфейса

По умолчанию Cyclone DDS **автоматически выбирает один** сетевой интерфейс по алгоритму:

1. Исключает loopback-интерфейсы
2. Исключает отключённые интерфейсы
3. Выбирает интерфейс с наивысшим приоритетом

Способы настройки:

В пункте 3 настройка Cyclone DDS будет рассмотрена более подробно и конкретно.

Через переменные окружения:

```
# Указание конкретного интерфейса
export CYCLONEDDS_URI="<General><NetworkInterfaceAddress>eth0</NetworkInterfaceAddress></General>"

# Использование всех интерфейсов
export CYCLONEDDS_URI="<General><NetworkInterfaceAddress>all</NetworkInterfaceAddress></General>"
```

Через конфигурационный файл:

```
<CycloneDDS>
  <Domain id="any">
    <General>
      <NetworkInterfaceAddress>eth0</NetworkInterfaceAddress>
    </General>
  </Domain>
</CycloneDDS>
```

6. Настройка Cyclone DDS для работы с роботом и симуляцией.

Настройка для симуляции

Cyclone DDS требует явного указания сетевого интерфейса для корректной работы в различных средах. Настройка осуществляется через переменную окружения CYCLONEDDS_URI.

1. Установите переменную окружения:

name="lo" - интерфейс, который используется для передачи информации. Нужно указать такой, какой Вы уже указали в конфигурационном файле mujoco - config.py.

Файл находится по адресу:

```
~/unitree_mujoco_mirea_olympiad/simulate_python/config.py
```

Поменяйте **name** и выполните команду:

```
export CYCLONEDDS_URI='<CycloneDDS><Domain><General><Interfaces>  
<NetworkInterface name="lo" priority="default" multicast="default"  
t" /></Interfaces></General></Domain></CycloneDDS>'
```

2. Перезапустите ROS Daemon для применения изменений:

```
ros2 daemon stop  
ros2 daemon start
```

Настройка для реального робота

1. Установите переменную окружения:

После подключения по Ethernet к роботу, необходимо определить активный сетевой интерфейс, по которому связаны компьютер и робот.

Выполните команду:

```
ip a
```

Появится список всех сетевых интерфейсов. В списке будет указано название интерфейса и его IP адрес, например: **eth0**.

name="eth0" - нужно поменять на тот который Вы узнали из вывода команды ip a.

Теперь выполните команду с измененным name:

```
export CYCLONEDDS_URI='<CycloneDDS><Domain><General><Interfaces>  
<NetworkInterface name="eth0" priority="default" multicast="default" /></Interfaces></General></Domain></CycloneDDS>'
```

2. Перезапустите ROS Daemon для применения изменений:

```
ros2 daemon stop  
ros2 daemon start
```

7. Trouble shooting, как проверить состояние сети/Daemon

Если Вы и робот находитесь в одной сети, но не видите топики /lowstate и /lowcmd, выполните следующие действия:

1. Выполните команду:

```
ros2 topic list
```

Если топики не отображаются, перейдите к следующему шагу.

2. Выполните команду:

```
cyclonedds ps
```

Если в списке присутствуют топики с префиксом rt/ и названиями lowstate и lowcmd, перезапустите демон ROS 2:

```
ros2 daemon stop  
ros2 daemon start
```

Если топики отсутствуют, перейдите к шагу 3.

3. Проверьте настройки сетевого интерфейса, который использует DDS:
Выполните команду:

```
echo $CYCLONEDDS_URI
```

Убедитесь, что интерфейс, через который Вы подключены к роботу или симуляции, совпадает с указанным в выводе команды.

Если интерфейс не совпадает, выполните команду:

```
export CYCLONEDDS_URI='<CycloneDDS><Domain><General><Interfaces>  
<NetworkInterface name="eth0" priority="default" multicast="default" /></Interfaces></General></Domain></CycloneDDS>'
```

заменяв **name="eth0"** на актуальное имя интерфейса.

После этого перезапустите демон ROS 2:

```
ros2 daemon stop  
ros2 daemon start
```

Если интерфейс указан верно, проверьте наличие опечаток (например, русские буквы или неверный символ). Ситуация, когда при правильно заданном сетевом интерфейсе отсутствуют DDS-топики и ROS-топики, исключена.

При работе с симуляцией убедитесь, что интерфейс в конфигурационном файле MuJoCo (config.py по пути `~/unitree_mujoco_mirea_olympiad/simulate_python/config.py`) совпадает с сетевым интерфейсом в переменной окружения CYCLONEDDS_URI. Интерфейс для связи с симуляцией может быть как внешним (ens33), так и внутренним (lo). **Главное, чтобы в конфигурационном файле MuJoCo и конфигурации DDS был указан один и тот же СУЩЕСТВУЮЩИЙ сетевой интерфейс.**