

Cộng Đồng ROS - Robot Operating System

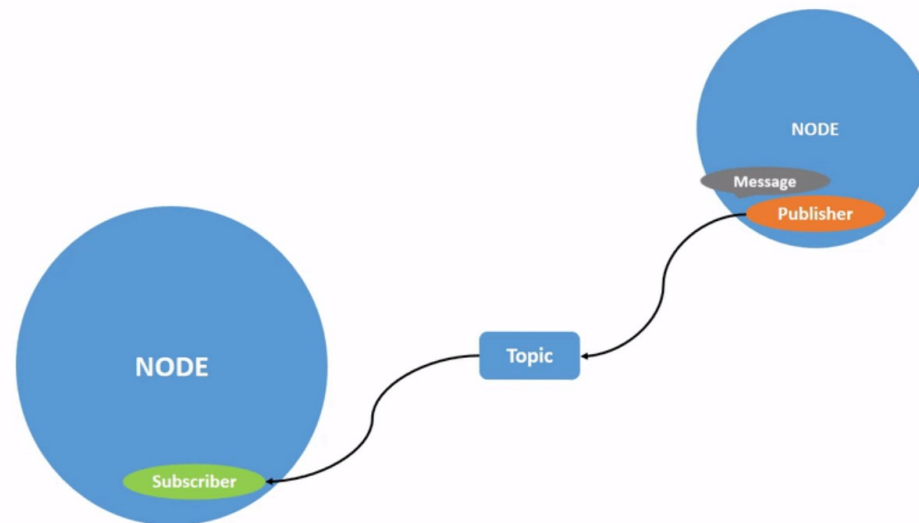
ROS2 TUTORIALS

For Basic Learning

1. TOPICS

1.1 Kiến thức cần nắm

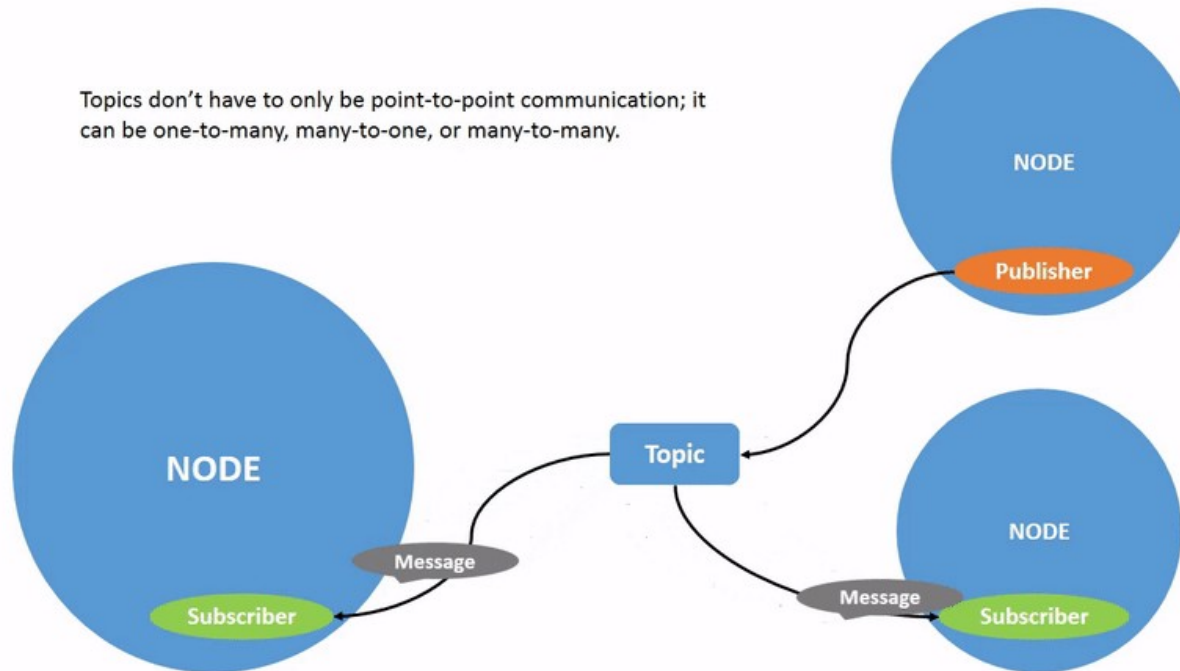
ROS2 tạo ra một hệ thống phức tạp tới nhiều mô đun các nodes. Topics là một yếu tố thiết yếu trong ROS graph, cái mà hoạt động như một đường nối cho các nodes để trao đổi messages.



1. TOPICS

Một node sẽ publish dữ liệu tới bất kì topics nào đó và đồng thời cũng có thể subscribers tới bất kỳ topics nào.

Topics don't have to only be point-to-point communication; it can be one-to-many, many-to-one, or many-to-many.



1. TOPICS

1.2 Thực hành

- Chạy node demo:

ros2 run turtlesim turtlesim_node

Ros2 run turtlesim turtle_teleop_key

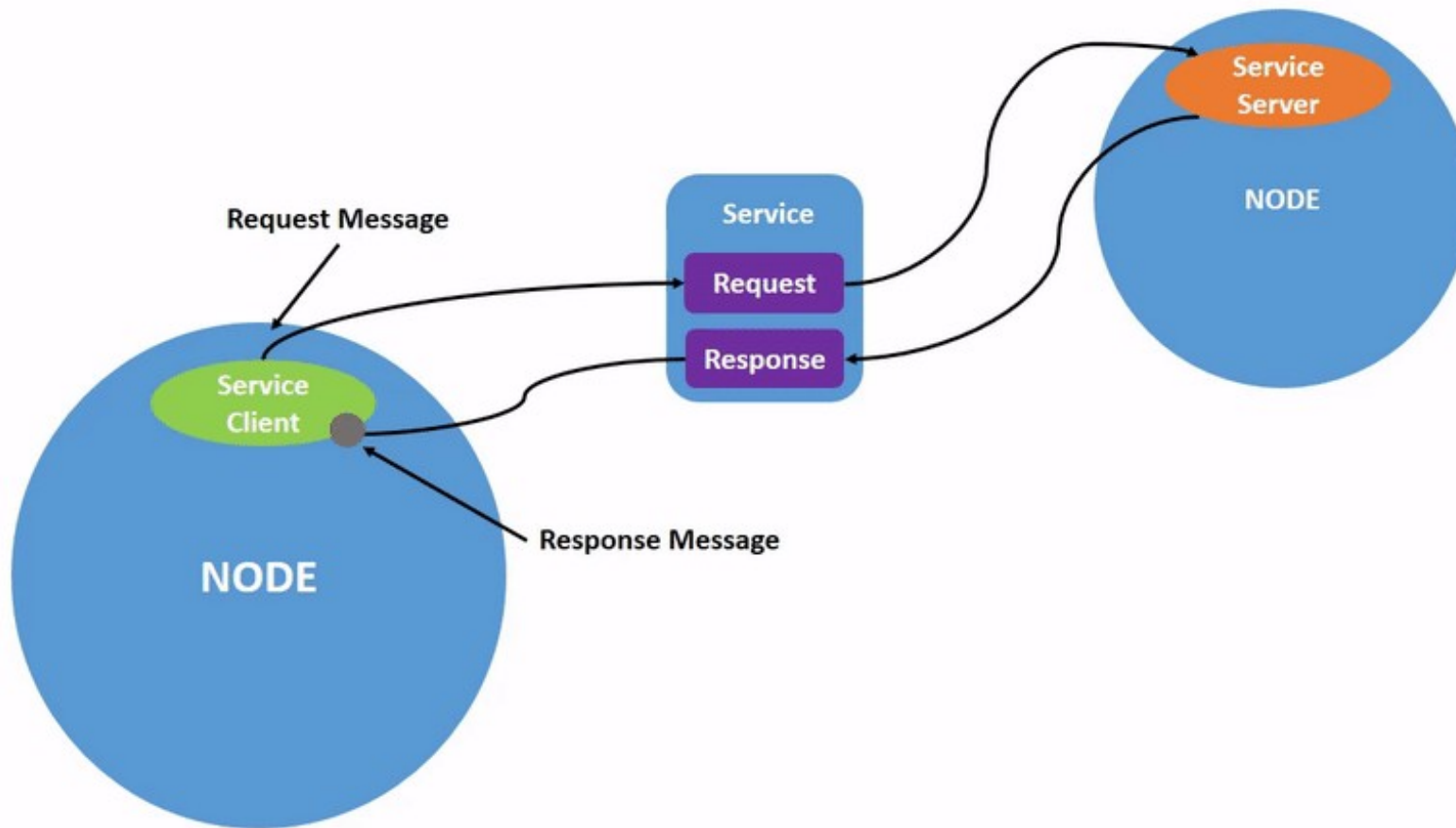
- rqt_graph
- ros2 topic pub
- ros2 topic list
- ros2 topic hz
- ros2 topic echo
- ros2 interface show <type_message>
- ros2 topic info

2. SERVICES

2.1 Kiến thức cần nắm

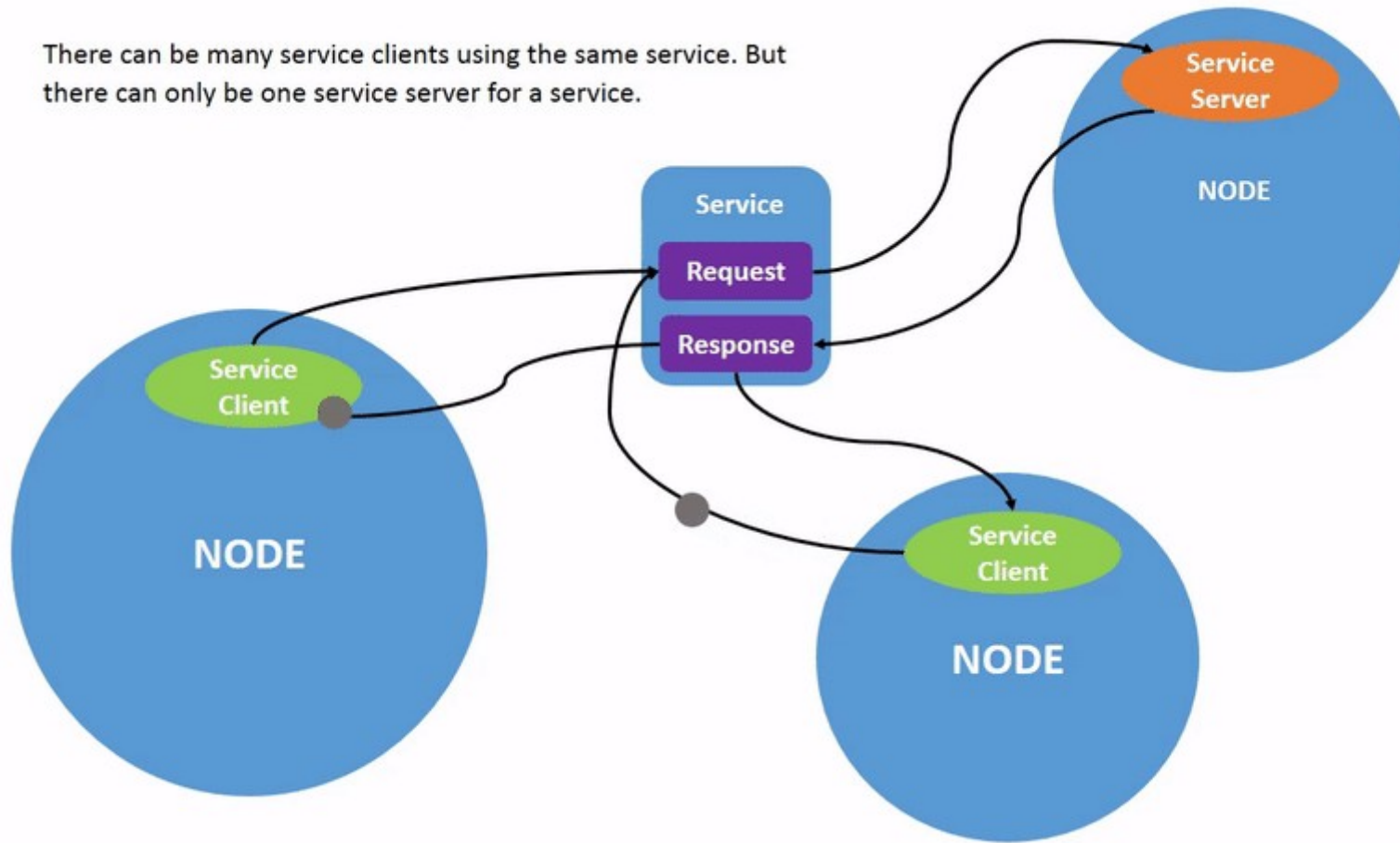
Services là một phương pháp khác của giao tiếp các nodes trên ROS graph. Trong khi topic cho phép nodes subscribe tới các luồng dữ liệu và có các cập nhật liên tục, Services chỉ cung cấp dữ liệu khi chúng được gọi chỉ định bởi một client.

2. SERVICES



2. SERVICES

There can be many service clients using the same service. But there can only be one service server for a service.



2. SERVICES

2.2 Thực hành

Chạy 2 nodes demo:

- ros2 service list
- ros2 service type
- ros2 service list -t
- ros2 service call

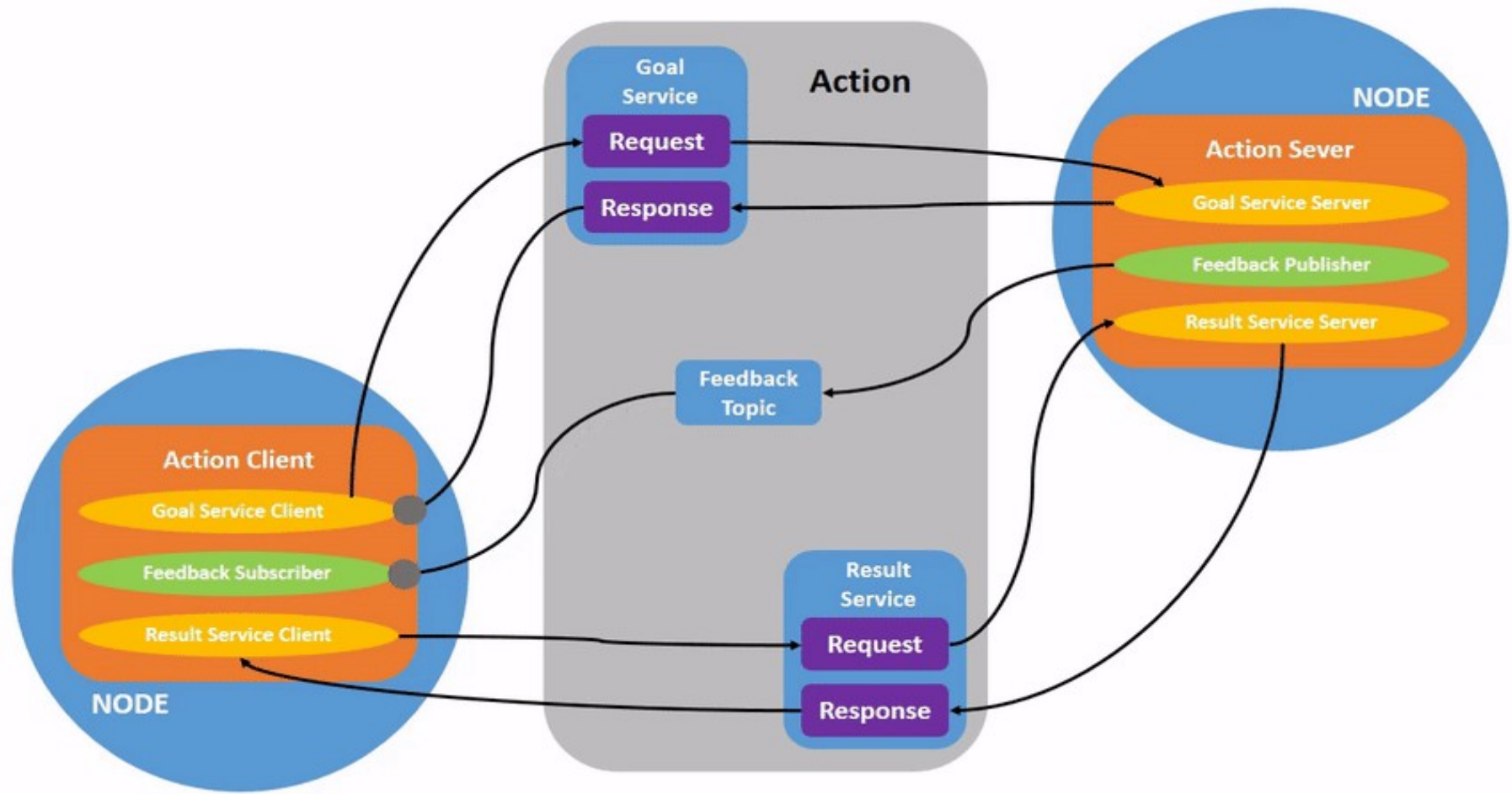


3. ACTIONS

Đặc điểm của Actions:

- Actions là một trong những kiểu giao tiếp trong ROS, được sử dụng cho các nhiệm vụ có thời gian thi hành lâu. Actions gồm 3 phần: a **goal**, a **result** và **feedback**.
- Actions được xây dựng trên các topics và services. Chức năng của nó tương tự như services, ngoại trừ actions có thể ưu tiên (và có thể hủy bỏ chúng trong khi đang thi hành). Chúng cũng cung cấp các phản hồi liên tục, ngược với service - chỉ trả về một phản hồi duy nhất.
- Actions sử dụng mô-đun **client - server**. **Action client** gửi **goal** tới một **action server**, cái node mà chứa thông tin về goal, sau đó trả về một luồng phản hồi như là một kết quả.

3. ACTIONS



3. ACTIONS

Các câu lệnh với Actions:

ros2 action list : in ra danh sách các actions đang hoạt động.

ros2 action list -t: in ra actions đang hoạt động và kiểu action.

ros2 action info /... : in ra thông tin về action.

ros2 action send_goal <action_name> <action_type> <values> --feedback

VD: `ros2 action send_goal /turtle1/rotate_absolute
turtlesim/action/RotateAbsolute "{theta: -1.57}" --feedback`

3. ACTIONS

3.3 Tạo một Actions

Các actions được định nghĩa theo kiểu .action:

Request

Result

Feedback

- **a request message:** được gửi từ một action client tới một action server cái mà tạo ra một new goal.
- **a result message:** được gửi từ một action server tới một action client khi mà một goal được thực hiện.
- **a feedback messages:** được gửi định kỳ từ một action server tới một action client với cập nhật về một goal.

3. ACTIONS

3.3 Tạo một Actions

Build một action:

Chỉnh sửa trong CmakeLists.txt

```
find_package(rosidl_default_generators REQUIRED)
```

```
rosidl_generate_interfaces(${PROJECT_NAME}  
  "action/Fibonacci.action"  
)
```

Với Fibonacci.action là action vừa được tạo.

Chỉnh sửa trong package.xml

```
<buildtool_depend>rosidl_default_generators</buildtool_depend>  
<depend>action_msgs</depend>  
<member_of_group>rosidl_interface_packages</member_of_group>
```

3. ACTIONS

3.4 Tạo Action Server và Action Client

4. ROS2DOCTOR

4.1 Background

Khi ROS2 setup của chúng ta đang chạy không như mong đợi, chúng ta có thể kiểm tra các cài đặt của nó với công cụ `ros2doctor`.

`Ros2doctor` kiểm tra tất cả các khía cạnh của ROS2, bao gồm platform, version, network, enviroment, running systems... và cảnh báo chúng ta về khả năng xảy ra của các lỗi cũng như là nguyên nhân.

4. ROS2DOCTOR

4.2 Thực hành `ros2doctor`

ros2 doctor: điều này sẽ kiểm tra tất cả các mô đun cài đặt, sau đó trả về cảnh báo, lỗi...