Introduction to ROS

ROS

- Robot Operating System
  - Not an operating system like mac/windows
  - A framework for robotic program communication and open source libraries

- Runs best on Linux (ubuntu)
  - ROS 1 Melodic (outdated) - Ubuntu 18
  - **ROS 1 Noetic - Ubuntu 20**
  - ROS 2 - Ubuntu 22/ window 11

Running Ubuntu

- Dedicated machine
  - A computer that runs ubuntu exclusively
- Dual Boot
  - A machine with both ubuntu and window/MAC
  - Segment the drive when installing
  - Requires reboot when switching between OS
- Virtual Machine
  - Simulating a different OS as a program in the native OS
  - Very resource hungry
  - May have problem when interfacing with hardware
  - Example: Oracle VirtualBox

Ubuntu Basics

- APT
  - Advanced Packaging Tools
  - Package installing tool for Linux
  - E.g.
    - *apt-get install* (instal package)
    - *apt-get update* (update package list version)
    - *apt-get upgrade* (upgrade package to the newest version according to package list)
- Sudo
  - Superuser do
  - “Run as admin”, if the command requires permission
  - E.g.
    - *sudo apt-get install “package”*
- cd
  - change directory

Instal ROS

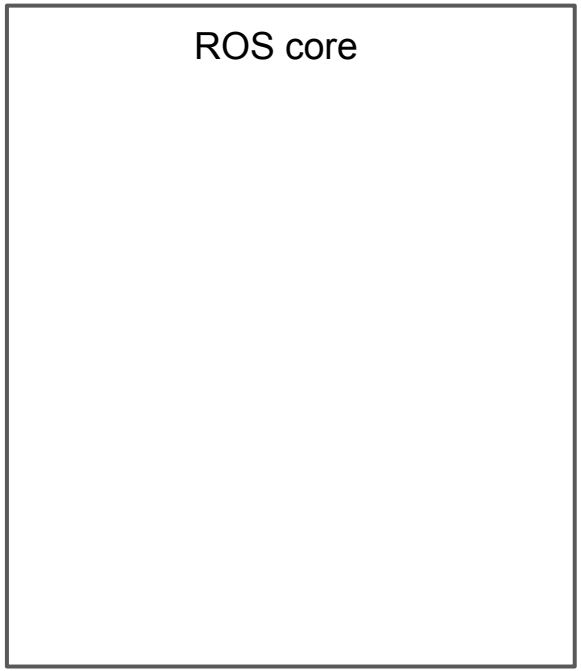
http://wiki.ros.org/noetic/Installation/Ubuntu

ROS basics - ROS Master

ROS Master

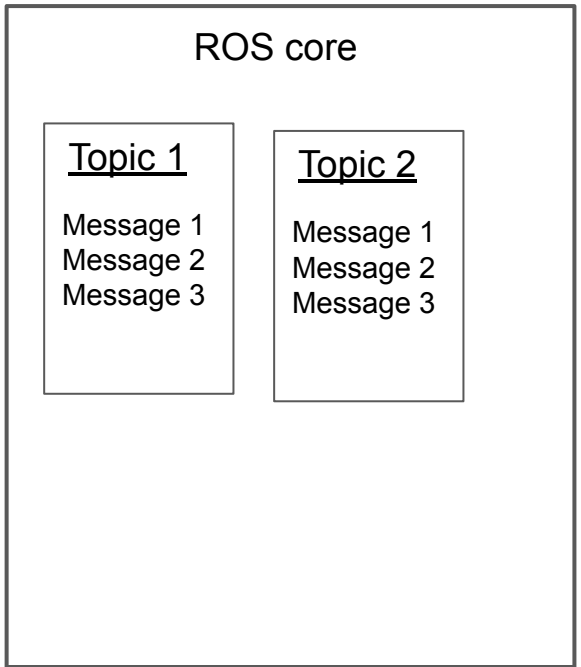
- The main program running the entire ROS system
- Handling Parameter server
- How to run:
  - *roscore*
  - Will automatically start when you run a ros package without existing rosore
    - Bad practice

ROS basics



ROS basics - Topic and Messages

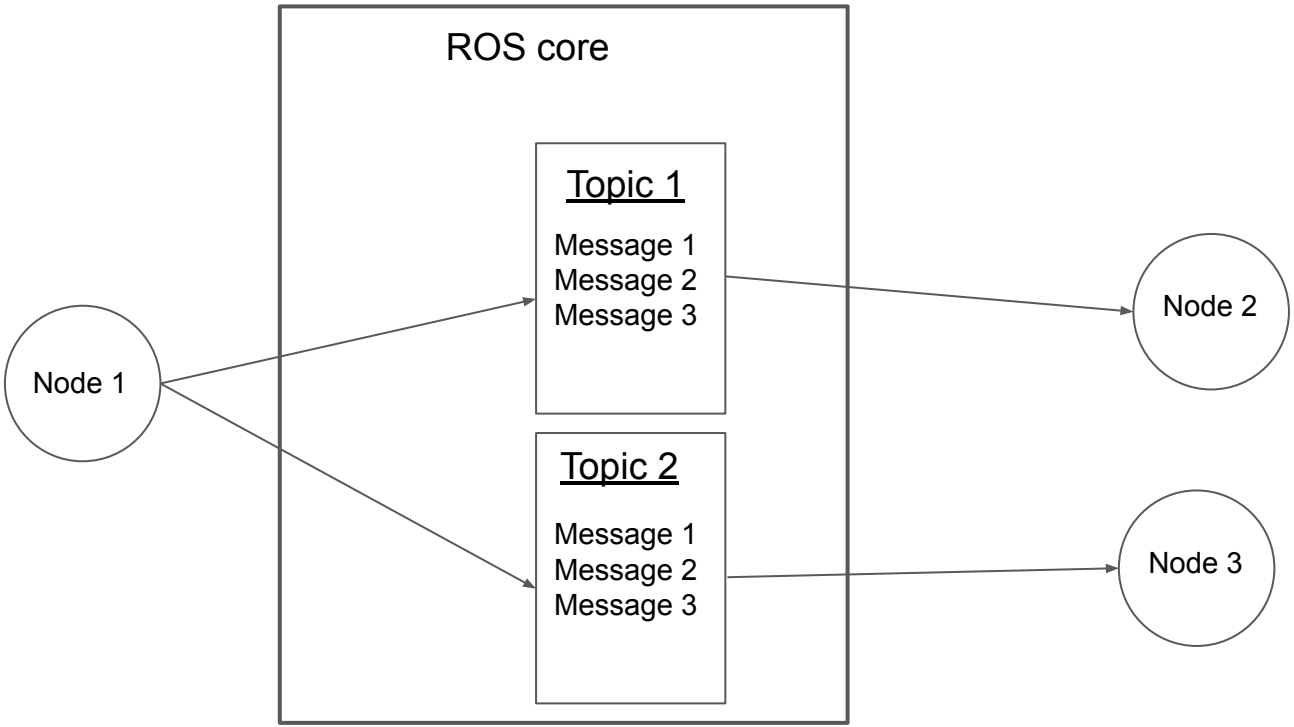
- Information is handled as messages
- messages are organized by topic
- Example 1:
  - Topic: camera/image
  - Message: RGB value of image at each timestep
- Example 2:
  - Topic: velocity_command
  - Message: velocity input to the robot
- Topic is like a postboard and message is like postnote, multiple program can read/ write messages to the same topic at the same time



ROS basics - Node

- Nodes are standalone programs that running within the ROS system
- Nodes are not inherently synchronized
- Node can receive message from ROS by subscribing to a topic
  - and Write message to ROS by publishing to a topic
- Example 1
  - Image processing node subscribes to image topic, process the image, and publish the label to a new topic
- Example 2
  - SLAM node subscribes to image topic and odometry topic and publish the map and localization

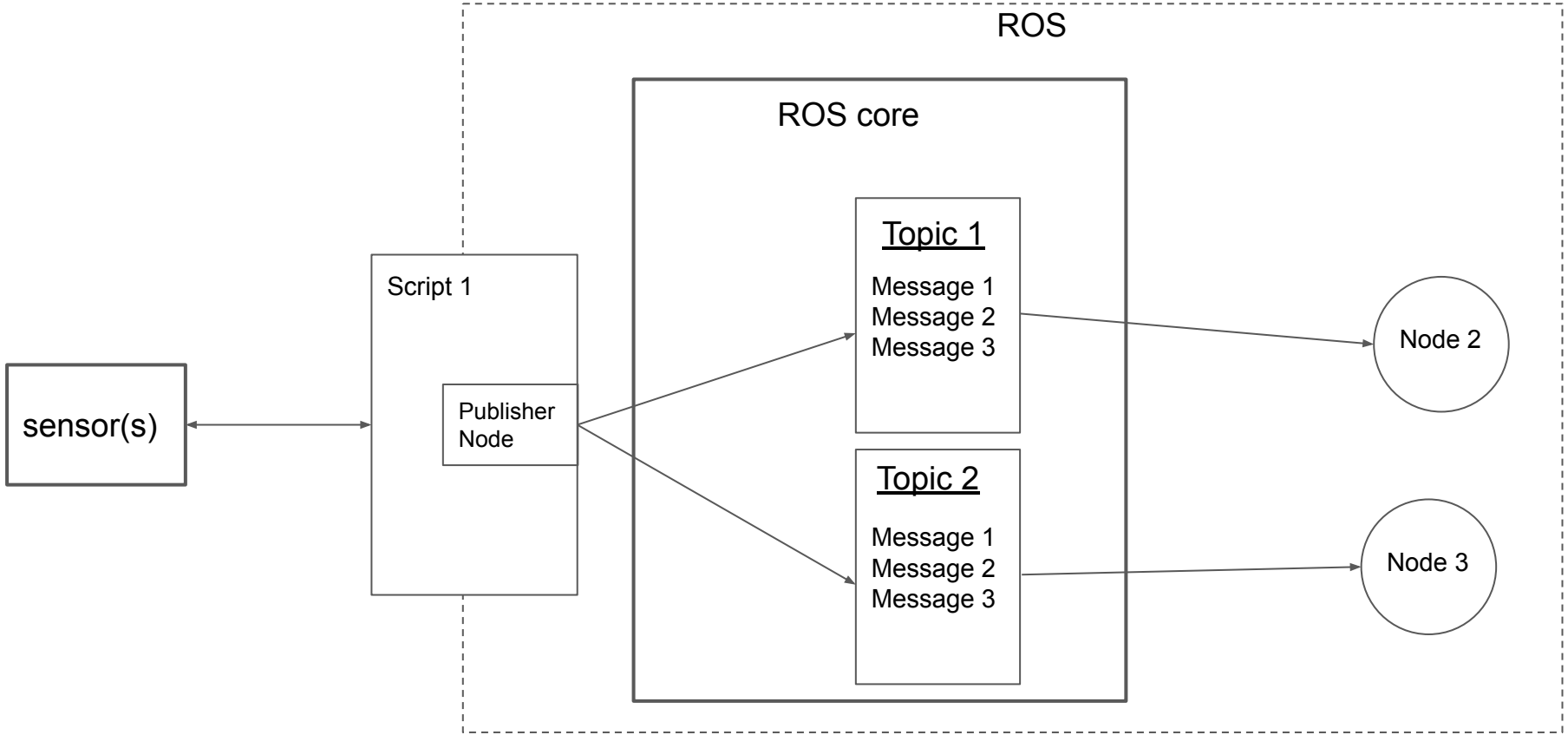
ROS basics - Node



ROS basics - Node

- Custom ROS node can be written in python or c++
- http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28python%29
- Subscriber nodes are event based
  - The script wait for message to arrive
  - Once message arrived, it triggers the callback function
  - Message can be processed in the callback function
- Publisher nodes usually publish at constant rate
  - Or at the end of callback function
  - Or whenever, doesn't matter
- You can do whatever else you want in the script

ROS basics - Node



ROS file system

- ROS packages are libraries that contain nodes, scripts, configuration files etc. for accomplishing a specific task
- Can be installed via binary (apt-get)
- Or build from source (catkin) (recommended)
- Check ros wiki for package documentation

ROS file system - Launch file

- ROS nodes are created by launch file
  - *roslaunch [package] [launch file]*
  - You can set parameters and which script to run in the launch file
- alternatively , rosrun can be used to launch a single script
  - For python, the script needs to be made executable by
    - *chmod +x [scriptname].py*
    - *rosrun [package] [scriptname].py*
  - For c++, the script needs to be added into the Cmake file, and compiled with *catkin_make* every time

Exercise - turtlesim

1. Install package
   - http://wiki.ros.org/turtlesim
2. Clone tutorial package to src
   - https://github.com/blue-ring-octopus/ros-tutorial
3. *catkin_make*
4. Create *move_turtle.py*
5. *chmod +x move_turtle.py*
6. Try remote master
   - *export ROS_MASTER_URI=[URI]*
   - *rosrun ros_tutorial move_turtle.py*

Exercise - turtlebot gazebo

- Install turtlebot dependency
  - https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/
  - Run 3.1.3 - 3.1.4 (from source)
- Install turtlebot gazebo simulation
  - https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#gazebo-simulation

Debugging Tools

- Rqt-graph
- TF-Tree
- Rostopic echo, list
- RVIZ