

Robot Operating System ROS

Una Introducción a ROS

 Qué es ROS?

 Comenzar con ROS

 Sistema de ficheros de ROS

 Conceptos gráficos de ROS

Qué es ROS?

- 🐢 abstracción de hardware y control de dispositivos de bajo nivel;
- 🐢 Creado por California-based Willow Garage, ahora mantenido por the Open Source Robotics Foundation (OSRF);
- 🐢 ROS = conectividad + herramientas + capacidades + ecosistema.

Qué es ROS?



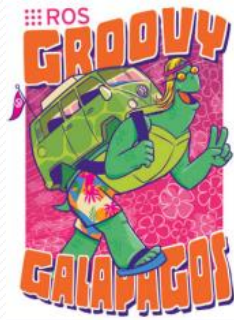
Jade



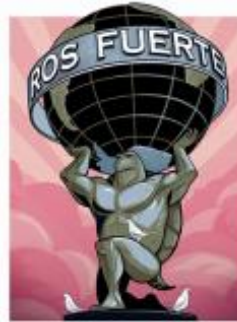
Indigo



Hydro



Groovy Galapagos



Fuerte Turtle



Electric



Diamondback



C Turtle



Box Turtle

Características princ. de ROS

- 🐢 abstracción de hardware y control de dispositivos de bajo nivel
- 🐢 independencia del lenguaje de programación
- 🐢 implementación de una amplia gama de herramientas y algoritmos de uso común
- 🐢 paso de mensajes entre procesos (independiente del sistema operativo)
- 🐢 gestión de paquetes estandarizados
- 🐢 conjunto útil de comandos y utilidades de shell con finalización de pestañas

ROS conceptos y componentes

Bibliotecas de clientes ROS

Principales:


- Python
- C++
- Lisp

Experimentales:

- Java (with Android support)
- Lua

Sistemas operativos compatibles

Sistemas operativos compatibles

  Ubuntu (14.04 LTS + ROS Indigo)

Experimentales



Arch



Mac OS X



Debian



OpenSuse



Fedora



Windows



Gentoo

Robots soportados



Nao



Willowgarage PR2



Baxter



Care-o-Bot



Toyota Helper



Gostai Jazz



Robonaut



Peoplebot



Kuka YouBot



Guardian



Husky A200



Summit



Turtlebot



Erratic



Qbo



AR.Drone



Miabot



AscTec



Lego NXT



Pioneer



SIA 10D

Muchos más en: <http://www.ros.org/wiki/Robots>

Sensores

🐢 1D/2D/3D range finders

- Sharp IR range finder
- Hokuyo laser scanners
- Sick lasers
- Microsoft Kinect
- Asus Xtion



Sensores

🐢 Cámaras

- monocular and stereo
- USB (uvc) and rewire
- video streaming (gstreamer)



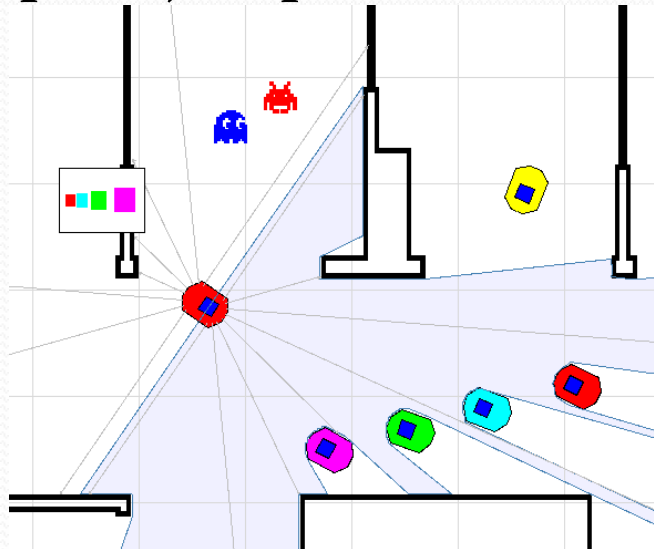
Sensores

- 🐢 Force/torque/touch sensors
- 🐢 Motion capture systems
- 🐢 Pose estimation (IMU/GPS)
- 🐢 Audio/Speech recognition
- 🐢 RFID
- 🐢 Sensor/actuator interfaces
 - Dynamixel
 - Phidgets
 - Arduino
 - Arbotix
 - Lego NXT
- 🐢 Y muchos más. . .



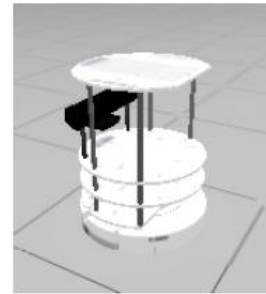
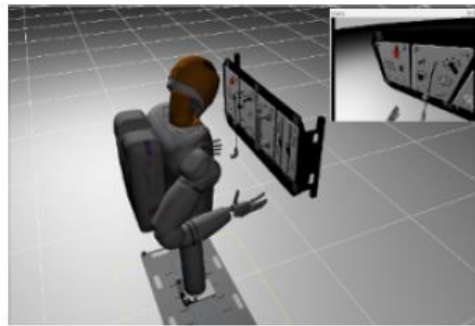
Simuladores - Stage

- 🐢 Stage es un simulador 2D para múltiples robots móviles (a gran escala)
- 🐢 Modelos para sensores (p. ej., láser, sonar) y actuadores (p. ej., pinza)
- 🐢 Modelos de objetos simples para manipulación (limitada)
- 🐢 Ningún modelo físico (por ejemplo, fricción, colisión, etc.)
- 🐢 Código abierto



Simuladores - Gazebo

- 🐢 Gazebo es un simulador 3D de múltiples robots en entornos realistas
- 🐢 Simulación realista de física/dinámica de cuerpos rígidos
- 🐢 Modelos para robots complejos, actuadores y sensores (cámaras, IMU)
- 🐢 Soporte proporcionado en parte por Open Source Robotics F.
- 🐢 Proyecto de código abierto



ROS filesystem - Visión general

Package

- Packages are the software organization unit of ROS code.
- Each package can contain libraries, executables, scripts, or other artifacts.
- Manifest: description (metadata) of a package, whose main role is to define dependencies between packages (package.xml)

Meta-packages

- Collection of packages forming a higher level library
- Previously called stacks. The concept of stacks was removed with catkin to simplify the growing code base and to support better distribution of packages.

Nodes

Nodes are processes which perform specific computations:

- 🐢 control robot wheel motors
- 🐢 acquire data from laser scanner
- 🐢 acquire images from camera
- 🐢 perform localisation
- 🐢 perform path planning
- 🐢 provide graphical visualisation of the system

Master

- 🐢 Master is the core node of ROS, called **roscore**
- 🐢 Acts as a **nameservice** for the Computation Graph
- 🐢 Stores topics and services registration information for ROS nodes
- 🐢 Nodes then establish connections as appropriate
- 🐢 Also makes callbacks to nodes when registration information changes
- 🐢 Allows nodes to dynamically create connections as new nodes are run

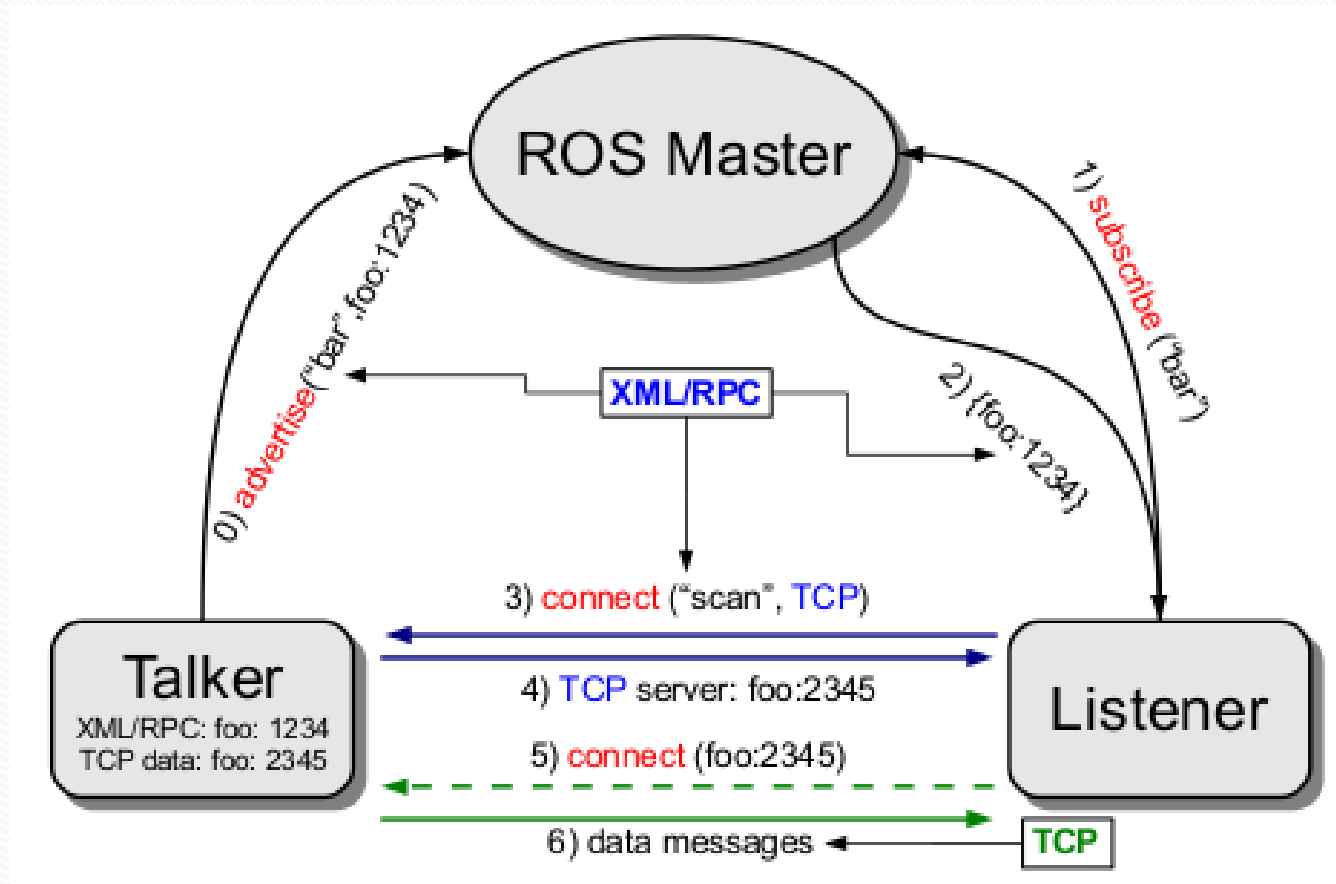
Messages

- 🐢 Messages are simply a data structure, consisting of typed fields
- 🐢 Standard primitive types (and nested arrays) are supported:
 - int8, 16, 32, 64
 - float32, 64
 - string
 - time
 - duration
 - array[]
 - For more information, go to <http://wiki.ros.org/msg>
- 🐢 Nodes communicate with each other by passing messages
- 🐢 Routed via a transport system with publish/subscribe semantics
- 🐢 When used with topics: *.msg (n:n)
- 🐢 When used with services: *.srv (1:1 { request + response})

Topics

- 🐢 A node sends out a message by publishing it to a given Topic
- 🐢 The topic type is defined by the message type publishing on it
- 🐢 A node requiring a certain type of data must subscribe to the appropriate Topic
- 🐢 Multiple publishers/subscribers to the same Topic are allowed
- 🐢 A single node may publish and/or subscribe to multiple Topics
- 🐢 Publishers and subscribers are generally unaware of each other's existence
- 🐢 Publish/subscribe model is a flexible paradigm (many-to-many, one-way transport)
- 🐢 There is no order of execution required

Topics - representación esquemática

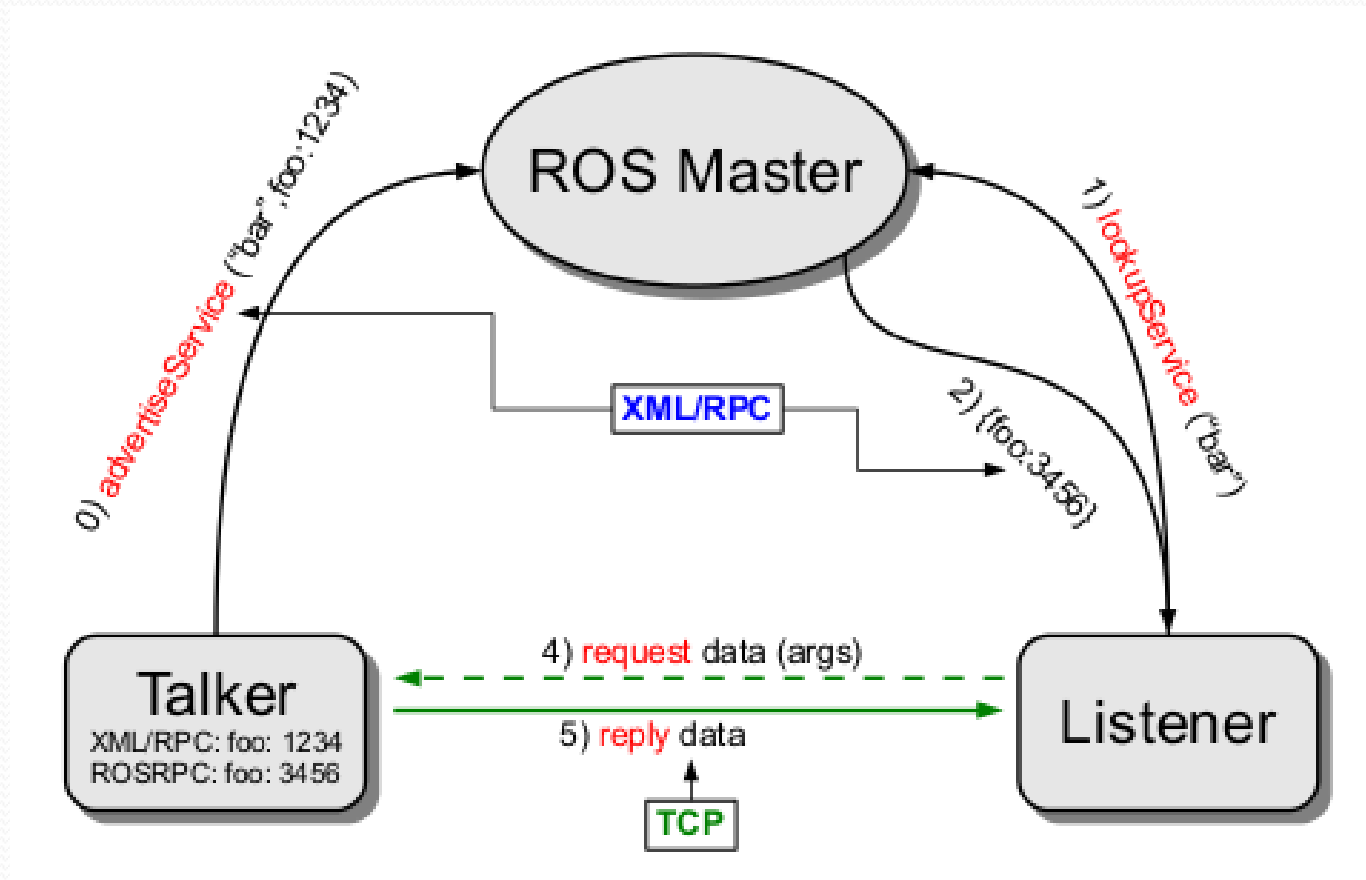


Xml/RPC: <http://en.wikipedia.org/wiki/XML-RPC>

Services

- 🐢 Publish/subscribe paradigm not appropriate for services
- 🐢 Services implement the request/reply functionality
- 🐢 Pair of message structures: one for request and one for reply
- 🐢 A node provider offers a service under a specific name
- 🐢 A client node uses the service by sending the request message and awaits for the reply
- 🐢 From the programmer perspective, works as a remote procedure call

Servicios - representación esquemática



roscore

roscore es una colección de nodos y programas que son requisitos previos de un sistema basado en ROS. Hay que tener un **roscore** ejecutándose para que los nodos ROS se comuniquen. Se lanza usando el comando **roscore**.

```
roscore http://demo-g580:11311/
sen@demo-g580:~$ roscore
... logging to /home/sen/.ros/log/2f9e6954-c4ac-11e3-8c62-c0143dce47b7/roslaunch-demo-g580-26541.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://demo-g580:35883/
ros_comm version 1.10.2

SUMMARY
=====

PARAMETERS
* /roscore
* /roscore

NODES

auto-starting new master
process[master]: started with pid [26555]
```

roslun

- 🐢 **roslun permite ejecutar un ejecutable en un paquete arbitrario sin conocer su ubicación**

`roslun package executable`

Eejmplo: `roslun cmd_vel_publisher cmd_vel_publisher_node`

- 🐢 **También es posible pasar parámetros**

`roslun package node _parameter:=value`

Ejemplo:

`roslun cmd_vel_publisher cmd_vel_publisher_node _Max_Constant_Vel:=0.5`

rostopic



Listado de comandos

- **rostopic kill** kill a running node
- **rostopic list** list active nodes
- **rostopic machine** list nodes running on a machines
- **rostopic ping** test connectivity to node
- **rostopic info** print information about node

rostopic

Listado de comandos admintidos

- **rostopic bw** display bandwidth used by topic
- **rostopic echo** print messages to screen
- **rostopic find** find topics by type
- **rostopic hz** display publishing rate of topic
- **rostopic info** print information about active topic
- **rostopic list** print informaion about active topics
- **rostopic pub** publish data to topic