

话题通信 妻子一个发布在话题一个信息，需要这个信息 从相应的话题 拿到信息  
这就算话题通信 发布订阅模式

服务通信 是 类似我访问一个网页 之后这个网页把相应的信息给你

参赛服务器 有一个容器 里面存储信息， 这里面的信息是共享的  
不同节点 都可以向这个容器 写入和读取数据

掌握各种通信机制

之后每种通信机制的理论模型

和代码实现

一些 ros 的基础操作

写几个例子

- 能够熟练介绍 ROS 中常用的通信机制
- 能够理解 ROS 中每种通信机制的理论模型
- 能够以代码的方式实现各种通信机制对应的案例
- 能够熟练使用 ROS 中的一些操作命令
- 能够独立完成相关实操案例

话题通信的场景

发布订阅模式

一个节点发布消息 一个节点订阅消息 通过节点耦合

机器人在执行导航功能，使用的传感器是激光雷达，机器人会采集激光雷达感知到的信息并计算，然后生成运动控制信息驱动机器人底盘运动。

在上述场景中，就不止一次使用到了话题通信。

- 以激光雷达信息的采集处理为例，在 ROS 中有一个节点需要时时的**发布当前雷达采集到的数据，导航模块中也有节点会订阅并解析雷达数据。**
- 再以运动消息的发布为例，**导航模块会根据传感器采集的数据时时的计算出运动控制信息并发布给底盘，底盘也可以有一个节点订阅运动信息并最终转换成控制电机的脉冲信号。**

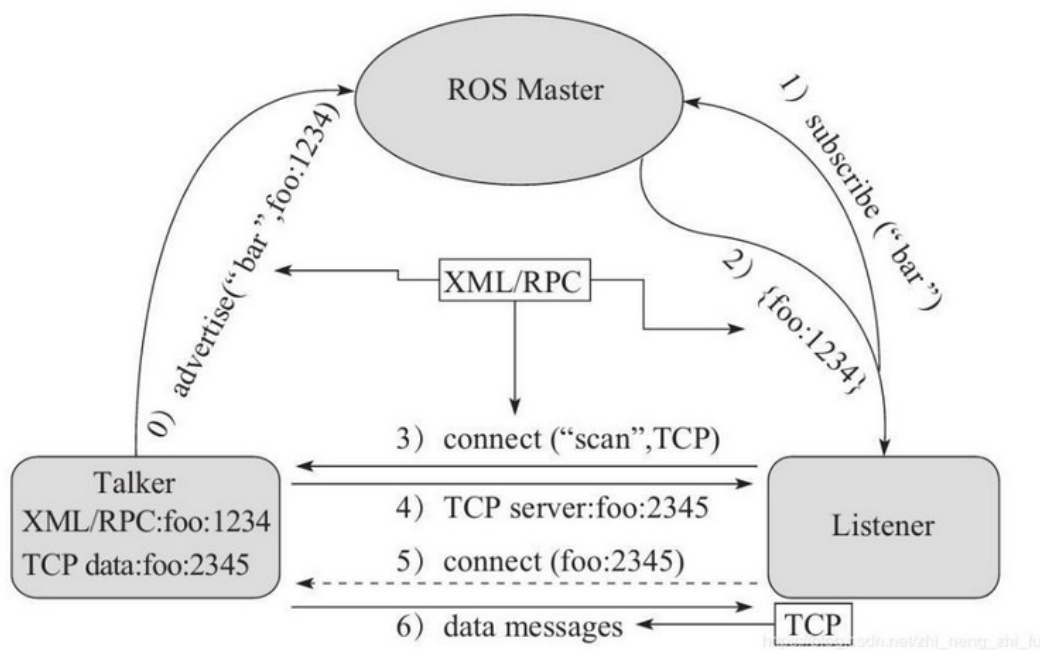
话题通信适用于 数据不停更新的摸索

通过发布 订阅 实现节点间的通信 一方发布信息 一方订阅信息

不断更新少逻辑处理的场景

## 话题通信的理论模型

ROS Master 负责保管 Talker 和 Listener 注册的信息，并匹配话题相同的 Talker 与 Listener，帮助 Talker 与 Listener 建立连接，连接建立后，Talker 可以发布消息，且发布的信息会被 Listener 订阅。



整个流程由以下步骤实现:

角色:

ROS Master : 管理者

Talker: 发布信息

Listener: 订阅 Talker 的信息

实现流程:

master 让 talker 和 listener 建立一个连接，让 2 者可以通信

第 0 talker 在 maseter 注册 自己的话题 自己的 RPC 地址

第 1 listener 在 maseter 主车 自己订阅的话题

第 2 mastet 比对发布者 订阅者的话题进行比对，如果话题一致 会把发布者的地质 给 lisetener

第 3listener 通过 RPC 地址与 talker 建立连接 访问 talker

第 4 talker 会响应 会包含 TCP 地址 把这些信息 给 listener

第 5 listener 通过 TCP 地址 与 listener 建立连接，listener 访问了 talker（根据 TCP 地址）

第 6 talker 通过 TCP 地址 向 listener 向 talker 发送信息

注意事项:

使用的协议是包含 RCP 和 TCP

0-4 是 RCP 建立连接

5-6 是 TCP 传递信息

0 和 1 两步没有先后关系

发布方和订阅方的数量, 发布方和订阅方 可以存在多个  
多个 talker 和多个 listener 可以同时存在多个

talker 和 listener 建立连接后, master 可以关闭

这些流程在 ros 已经封装, 在后续可以直接调用

掉包:

编写 talker 和 listener

设置话题

关注消息的发布