

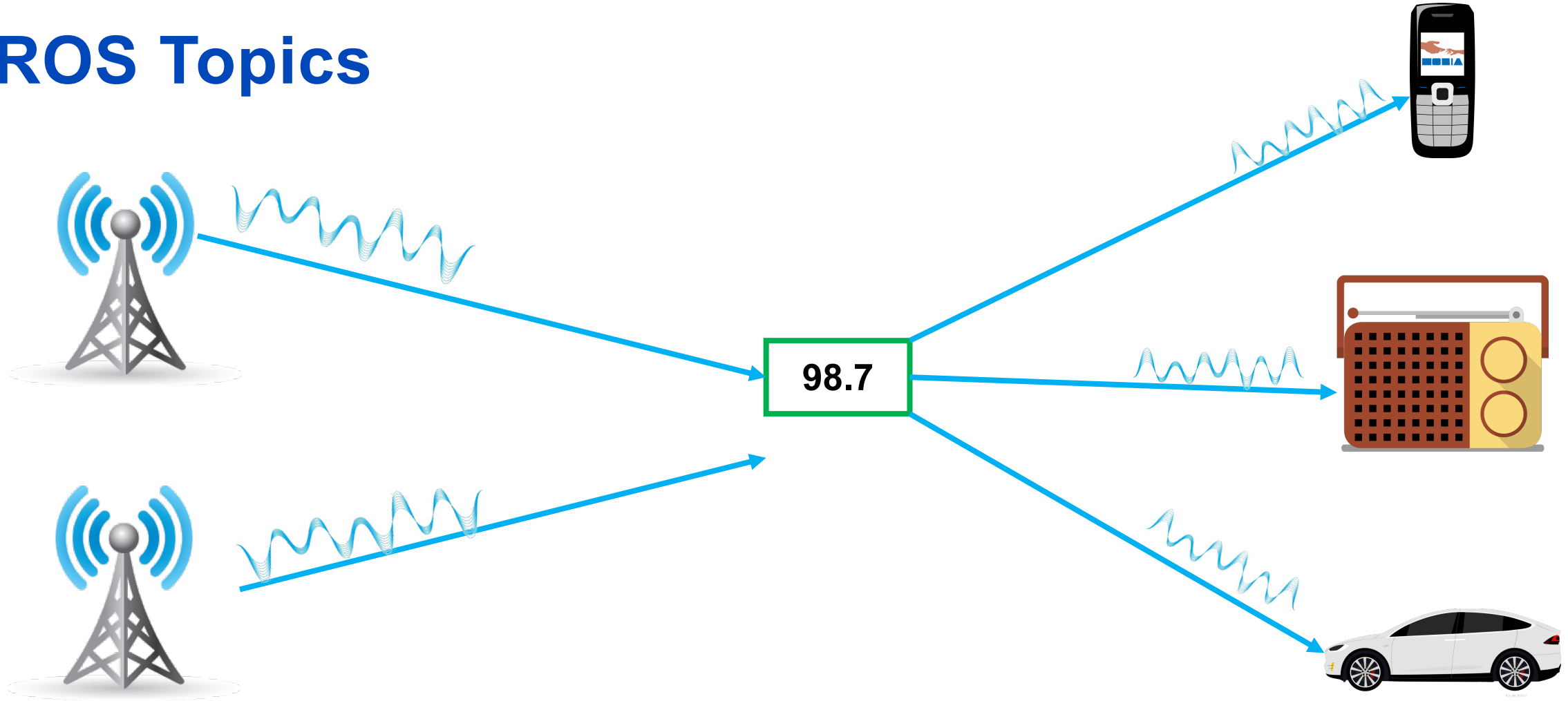


جامعة خليفة
Khalifa University

Communicate with ROS Topic

JUNE-2021

ROS Topics



ROS Topics

- A topic is a named bus over which nodes exchange messages
- Unidirectional data stream (publisher/subscriber)
- Anonymous
- A topic has a message type
- Can be written in python, C++, Directly inside ROS nodes
- The ROS master helps nodes finding needed topics
- A node can have many publishers/subscribers for many different topics

ROS Topics

What is Topic?

Create a Python Publisher

Create a Python Subscriber

Use Anonymous Nodes to Launch Multiple
Publisher/Subscriber

Debug Topics With Command Line Tools

Visualize Topics With `rqt_graph`

Experiment on Topics With `Turtlesim`

Create a Python Publisher

```
$ cd catkin_ws/src/my_robot_tutorials/scripts/  
$ ls  
$ touch robot_news_radio_transmitter.py  
$ chmod +x robot_news_radio_transmitter.py  
$ gedit robot_news_radio_transmitter.py
```

```
ros@ros-vm: ~/catkin_ws/src/my_robot_tutorials/scripts  
ros@ros-vm:~$ cd catkin_ws/src/my_robot_tutorials/  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials$ cd scripts/  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls  
my_first_node.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ touch robot_news_radio_tr  
er.py
```

```
ros@ros-vm: ~/catkin_ws/src/my_robot_tutorials/scripts  
ros@ros-vm:~$ cd catkin_ws/src/my_robot_tutorials/  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials$ cd scripts/  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls  
my_first_node.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ touch robot_news_radio_transmitt  
er.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ chmod +x robot_news_radio_transm  
itter.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls  
my_first_node.py  robot_news_radio_transmitter.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ vim robot_news_radio_transmitter  
.py
```

```
service = rospy.Service("/add_two_ints", AddTwoInts)
```

```
ros@ros-vm: ~/catkin_ws/src/my_robot_tutorials/scripts
ros@ros-vm:~$ cd catkin_ws/src/my_robot_tutorials/
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials$ cd scripts/
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls
my_first_node.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ touch robot_news_radio_transmitter.py
```

```
ros@ros-vm: ~/catkin_ws/src/my_robot_tutorials/scripts
ros@ros-vm:~$ cd catkin_ws/src/my_robot_tutorials/
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials$ cd scripts/
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls
my_first_node.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ touch robot_news_radio_transmitter.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ chmod +x robot_news_radio_transmitter.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls
my_first_node.py  robot_news_radio_transmitter.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ vim robot_news_radio_transmitter.py
```

Now edit the file

```
#!/usr/bin/env python3
```

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String

if __name__ == '__main__':

    rospy.init_node('robot_news_radio_transmitter')

    pub = rospy.Publisher("/robot_news_radio", String, queue_size=10)

    rate = rospy.Rate(2)

    while not rospy.is_shutdown():
        msg = String()
        msg.data = "Hi, this is Dan from the Robot News Radio !"
        pub.publish(msg)
        rate.sleep()

    rospy.loginfo("Node was stopped")
```

~

~

Terminal

```
$ python3 robot_news_radio_transmitter.py
```

```
$roscore
```

```
$ cd  
$ rostopic list
```

```
ros@ros-vm:~/catkin_ws/src/my_robot_tutorial...  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorial/scripts$ cd  
ros@ros-vm:~$ rostopic list  
/robot_news_radio  
/rosout  
/rosout_agg  
ros@ros-vm:~$
```

```
/rosout_agg  
ros@ros-vm:~$ rostopic echo /robot_news_radio  
data: "Hi, this is Dan from the Robot News Radio. I
```

```
$ rosnode list  
/robot_news_radio_transmitter  
/rosout
```

Create a Python Service Client

```
$ cd catkin_ws/src/my_robot_tutorials/scripts/  
$ ls  
$ touch add_two_ints_client.py  
$ chmod +x add_two_ints_client.py  
$ gedit add_two_ints_client.py
```

Now edit the file

```
ros@ros-vm:~$ cd catkin_ws/src/my_robot_tutorials/scripts/  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls  
add_two_ints_server.py  number_counter.py  robot_news_radio_transmitter.py  
my_first_node.py      number_publisher.py  smartphone.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ touch add_two_ints_client.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ chmod +x add_two_ints_  
chmod: cannot access 'add_two_ints_': No such file or directory  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ chmod +x add_two_ints_client.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls  
add_two_ints_client.py  number_counter.py  smartphone.py  
add_two_ints_server.py  number_publisher.py  
my_first_node.py      robot_news_radio_transmitter.py  
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ vim a
```

```
#!/usr/bin/env python3  
Import rospy  
From rospy_tutorials.srv import AddTwoInts  
If __name__ == '__main__':  
    rospy.init_node("add_two_ints_server")  
    rospy.loginfo("Add two ints server node created")  
    service = rospy.Service("/add_two_ints", AddTwoInts)
```

Debug Services with Command Line Tools

Experiment on Services with Turtlesim

```
snailab@snailab-System-Product-Name: ~
snailab@snailab-System-Product-Name:~$ rosrun turtlesim turtlesim_node
[ INFO] [1624745532.034452019]: Starting turtlesim with node name /turtlesim
[ INFO] [1624745532.037133888]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
[ INFO] [1624745722.292990489]: Resetting turtlesim.
[ INFO] [1624745722.314960740]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
[ INFO] [1624745737.957828696]: Resetting turtlesim.
[ INFO] [1624745737.978230296]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
[ INFO] [1624745746.853944659]: Resetting turtlesim.
[ INFO] [1624745746.875863151]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
snailab@snailab-System-Product-Name:~$ rosservice

roscore http://snailab-System-Product-Name:11311/81x26
unch-snailab-System-Product-Name-13219.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://snailab-System-Product-Name:40599/
ros_comm version 1.15.11

SUMMARY
=====

PARAMETERS
* /roscpp: noetic
* /rosversion: 1.15.11

NODES

auto-starting new master
process[roscpp]: started with pid [13227]
ROS_MASTER_URI=http://snailab-System-Product-Name:11311/

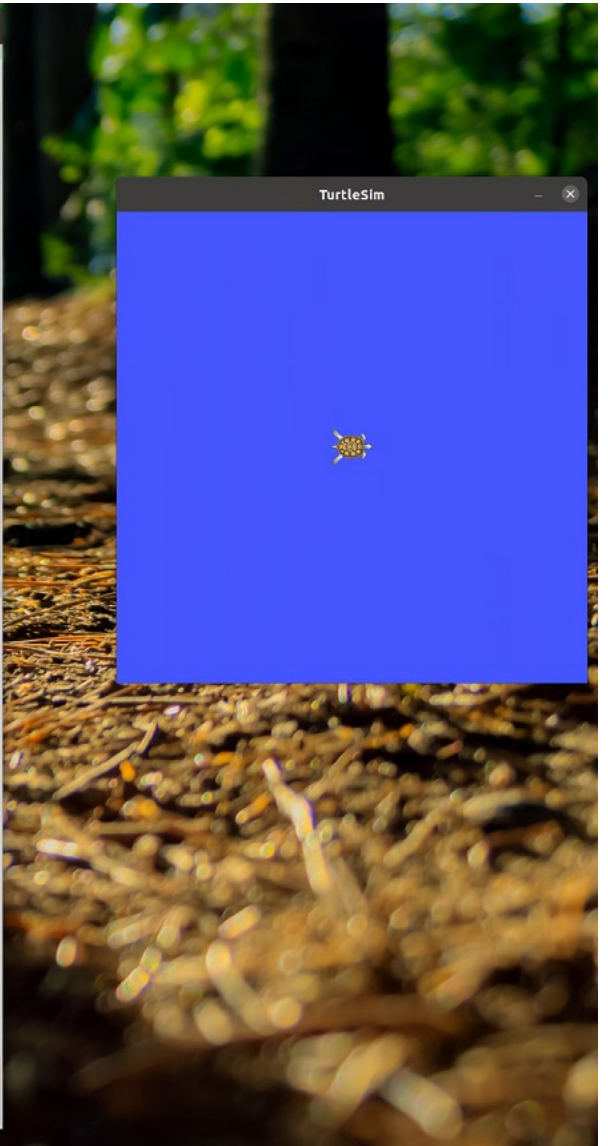
setting /run_id to 2ac7011e-d6c1-11eb-bf04-df8479f94907
process[roscpp-1]: started with pid [13237]
started core service [/roscpp]

angular: 2.0"

snailab@snailab-System-Product-Name:~$ rosservice list
/clear
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/spawn
/teleop_turtle/get_loggers
/teleop_turtle/set_logger_level
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/get_loggers
/turtlesim/set_logger_level
snailab@snailab-System-Product-Name:~$ rosservice call /turtle1/teleport_absolute
Usage: rosservice call /service [args...]

rosservice: error: Please specify service arguments
snailab@snailab-System-Product-Name:~$ rosservice call /r
/reset
/rosout/get_loggers /rosout/set_logger_level
snailab@snailab-System-Product-Name:~$ rosservice call /reset

snailab@snailab-System-Product-Name:~$
```



Experiment on Services with Turtlesim