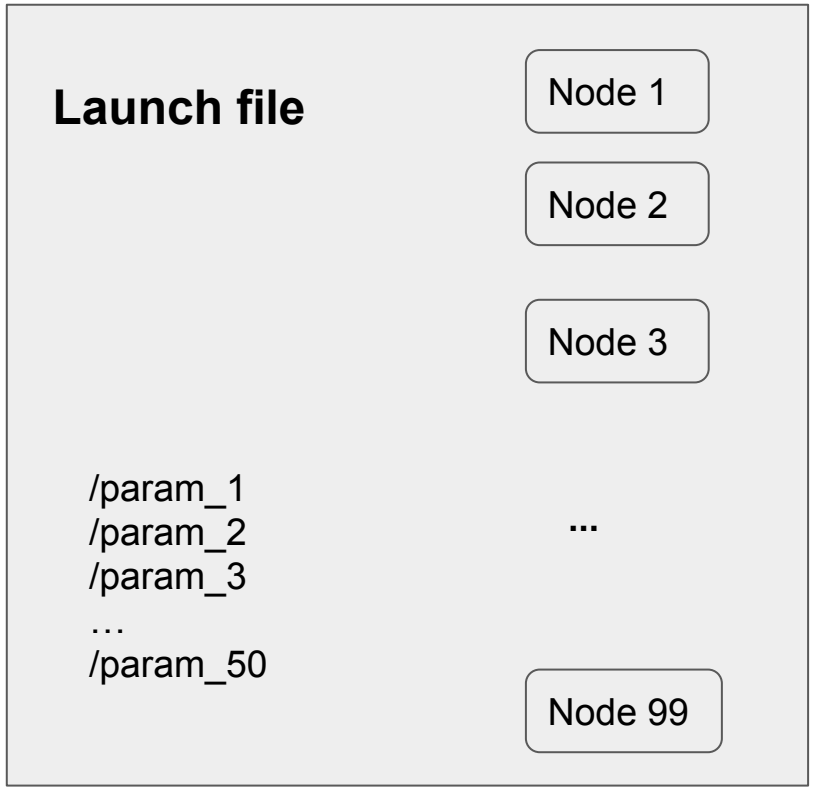
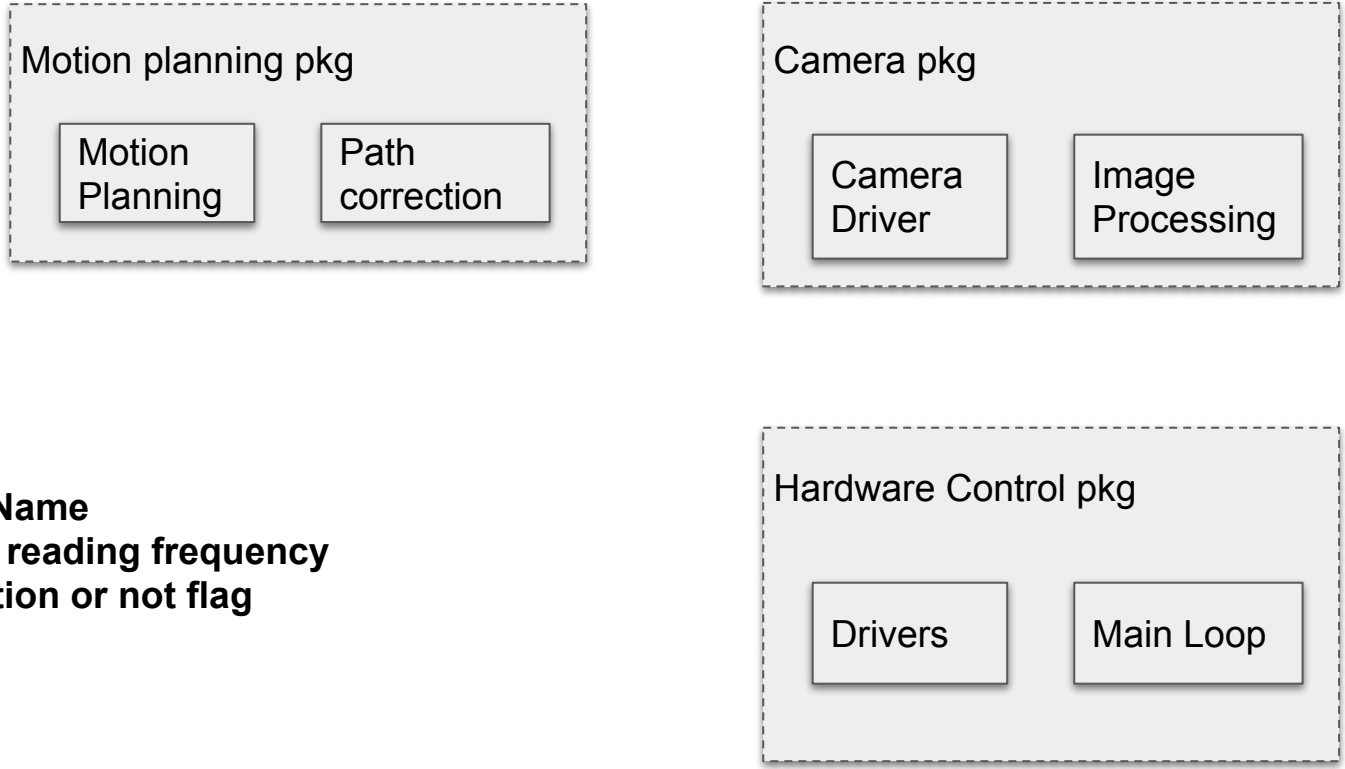
Make Your Application Scalable With ROS Params and Launch Files

IRR S2019

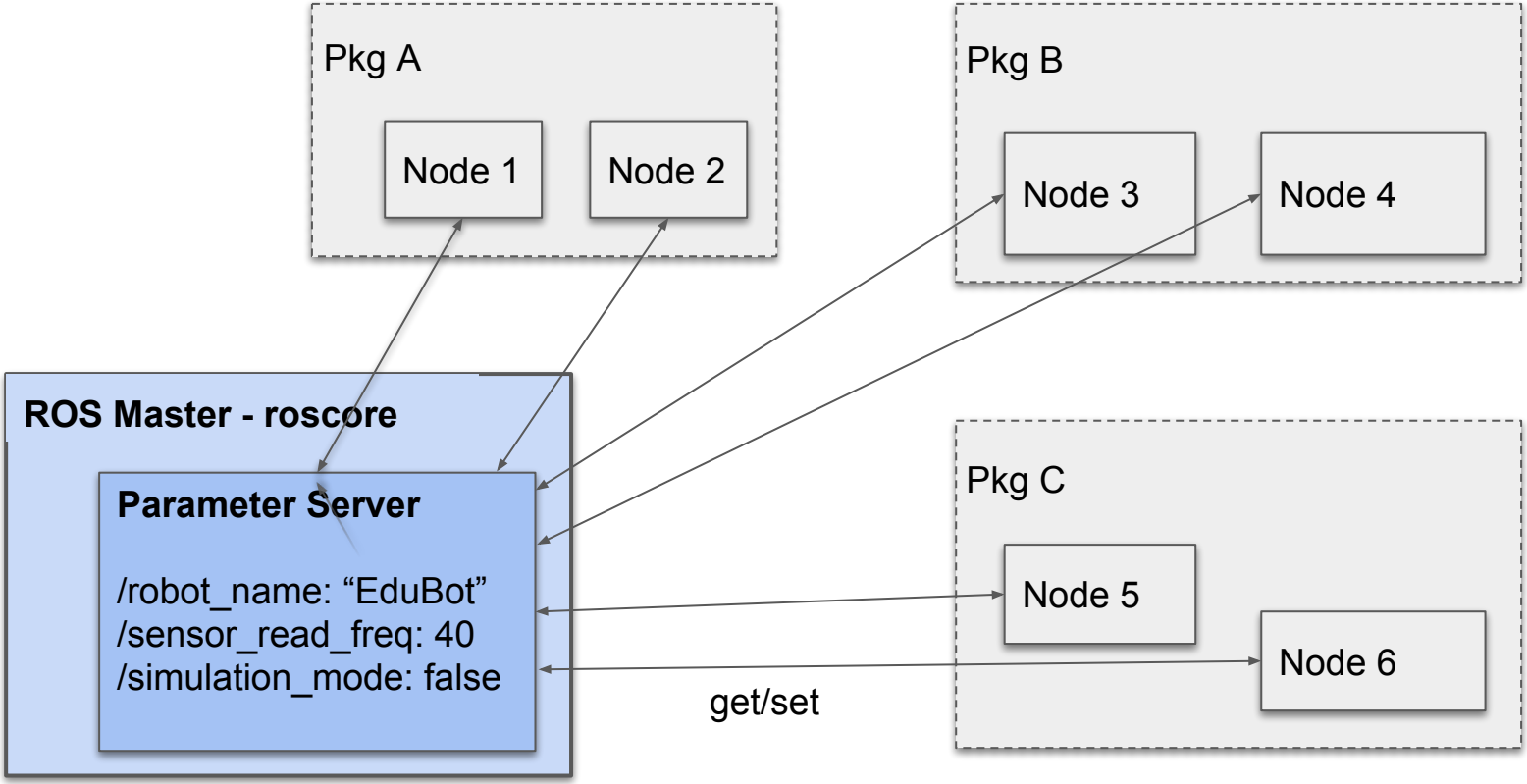


ROS Parameters



- **Robot Name**
- **Sensor reading frequency**
- **Simulation or not flag**
- **...**

ROS Parameters



ROS Parameters

Parameter Server: dictionary inside ROS master, accessible globally

ROS parameter: a shared variable stored inside the parameter server

ROS parameter types

- Boolean
- Int
- Double
- String
- Lists
- ...

Manipulate Parameters with ROS Commands

```
cjchung@Robofest:~/catkin_ws$ rosparam -h
rosparam is a command-line tool for getting, setting, and deleting parameters fr
om the ROS Parameter Server.

Commands:
        rosparam set    set parameter
        rosparam get    get parameter
        rosparam load   load parameters from file
        rosparam dump   dump parameters to file
        rosparam delete delete parameter
        rosparam list   list parameter names
```

```
cjchung@Robofest:~/catkin_ws$ rosparam list
/rosdistro
/roslaunch/uris/host_robofest__42501
/rosversion
/run_id
```

```
cjchung@Robofest:~/catkin_ws$ rosparam set /robot_name "EduBot"
cjchung@Robofest:~/catkin_ws$ rosparam list
/robot_name
/rosdistro
/roslaunch/uris/host_robofest__42501
/rosversion
/run_id
cjchung@Robofest:~/catkin_ws$ rosparam set /sensor_read_freq 40
cjchung@Robofest:~/catkin_ws$ rosparam set /simulation_mode false
cjchung@Robofest:~/catkin_ws$ rosparam list
/robot_name
/rosdistro
/roslaunch/uris/host_robofest__42501
/rosversion
/run_id
/sensor_read_freq
/simulation_mode
```

```
cjchung@Robofest:~/catkin_ws$ rosparam get /robot_name
EduBot
cjchung@Robofest:~/catkin_ws$ rosparam get /sensor_read_freq
40
cjchung@Robofest:~/catkin_ws$ rosparam get /simulation_mode
false
```

Handle Parameters with C++

1. First run roscore, then **`$ rosparam set /news_topic1_pub_freq 0.5`**
2. Modify `news_publisher_yourname.cpp` file as the following

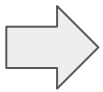
To read a parameter in C++

```
#include <ros/ros.h>
#include <std_msgs/String.h>

int main (int argc, char **argv)
{
    ros::init(argc, argv, "news_publisher_cj");
    ros::NodeHandle nh;

    ros::Publisher pub = nh.advertise<std_msgs::String>("/news_topic1", 10);

    double pub_freq = 1; // defualt is 1 Hz
    nh.getParam("news_topic1_pub_freq", pub_freq);
    ros::Rate rate(pub_freq);    // ros::Rate rate(4);

    while (ros::ok()) {
        std_msgs:: String msg;
        msg.data = "topic1 news by cj in c++";
        pub.publish(msg);
        rate.sleep();
    }
```

Run `news_publisher_yourname.cpp` and `news_subscriber_chris.cpp` after `$ catkin_make`

```
cjchung@Robofest:~$ rosrun my_ros_tutorials news_publisher_cj
```

Now this message will be displayed every 2 (1/0.5) seconds

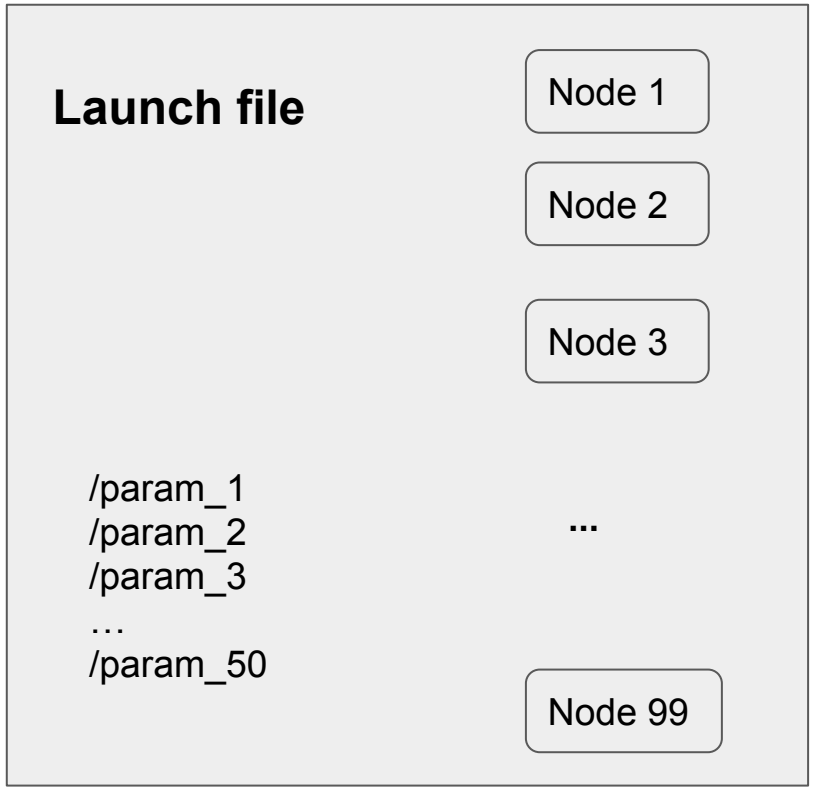


```
cjchung@Robofest: ~
cjchung@Robofest:~$ rosrun my_ros_tutorials news_subscriber_chris
[ INFO] [1548257301.650850971]: Message received:  topic1 news by cj in c++
[ INFO] [1548257301.900635941]: Message received:  topic1 news by cj in c++
[ INFO] [1548257302.150569943]: Message received:  topic1 news by cj in c++
[ INFO] [1548257302.400576305]: Message received:  topic1 news by cj in c++
[ INFO] [1548257302.650615632]: Message received:  topic1 news by cj in c++
[ INFO] [1548257302.900564683]: Message received:  topic1 news by cj in c++
[ INFO] [1548257303.150598038]: Message received:  topic1 news by cj in c++
[ INFO] [1548257303.400668780]: Message received:  topic1 news by cj in c++
[ INFO] [1548257303.650672395]: Message received:  topic1 news by cj in c++
[ INFO] [1548257303.900673774]: Message received:  topic1 news by cj in c++
```

What is a ROS Launch File?

Suppose you have 99 nodes and 50 parameters to set.

Is there a better way than starting/setting one by one as we did before?

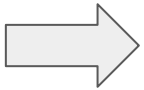


Let's first test HW1 using a launch file with 4 nodes

1. Create a "**launch**" director in the package, my_ros_tutorials
2. Create a file, "hw1.launch"

```
<launch>
    <node name="news_publisher_cj" pkg="my_ros_tutorials" type="news_publisher_cj" />
    <node name="news_subscriber_chris" pkg="my_ros_tutorials" type="news_subscriber_chris"
          output="screen" launch-prefix="gnome-terminal -e" />
    <node name="news_changer" pkg="my_ros_tutorials" type="news_changer" />
    <node name="news_subscriber_xx" pkg="my_ros_tutorials" type="news_subscriber_xx"
          output="screen" launch-prefix="gnome-terminal -e" />
</launch>
```

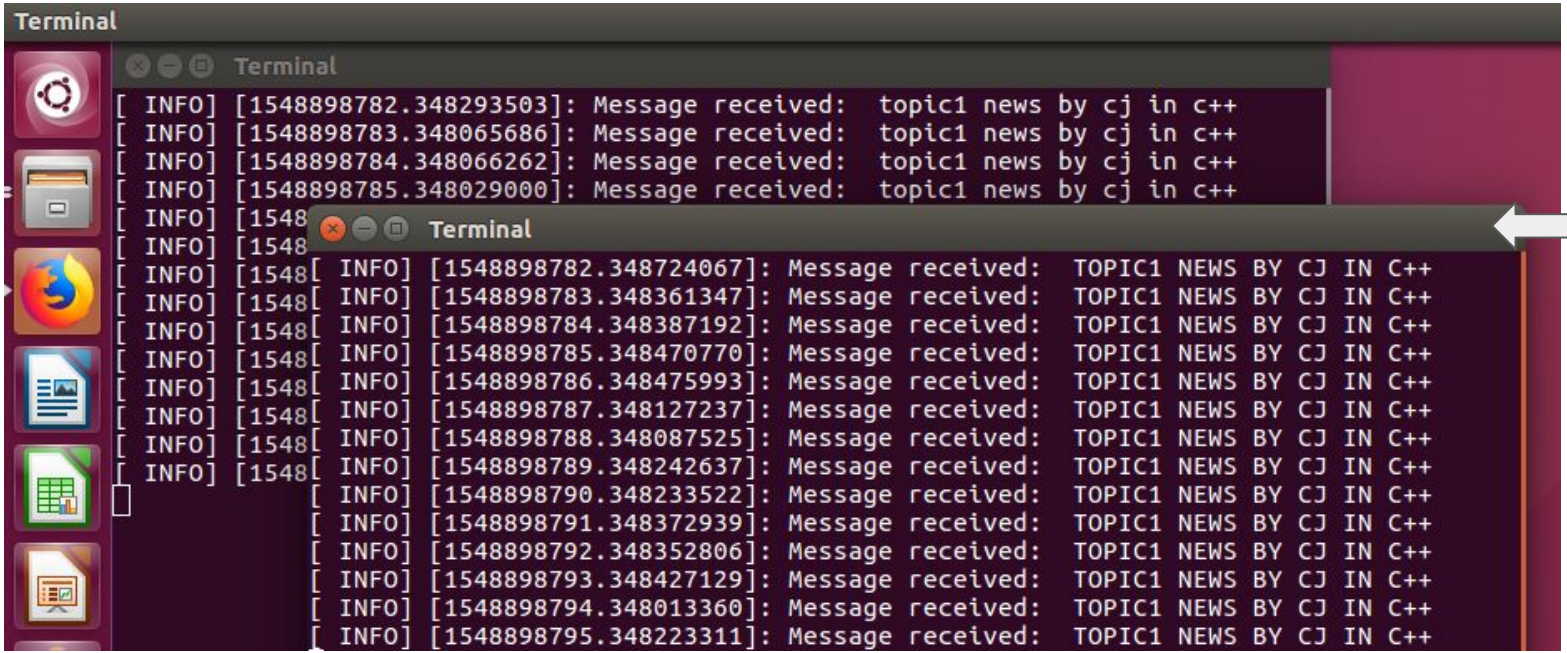
3. $ roslaunch hw1.launch



Two terminals will be started. Need to move the top one to see both.

Note that $ roscore is automatically started.

```
cjchung@Robofest:~/catkin_ws/src/my_ros_tutorials/launch$ roslaunch my_ros
_tutorials hw1.launch
```



Overlapped. Need to move this Terminal

How to create a Launch file to set /news_topic1_pub_freq with 1.5

```
<launch>
    <param name="/news_topic1_pub_freq" type="double" value="1.5" />

    <node ...

</launch>
```

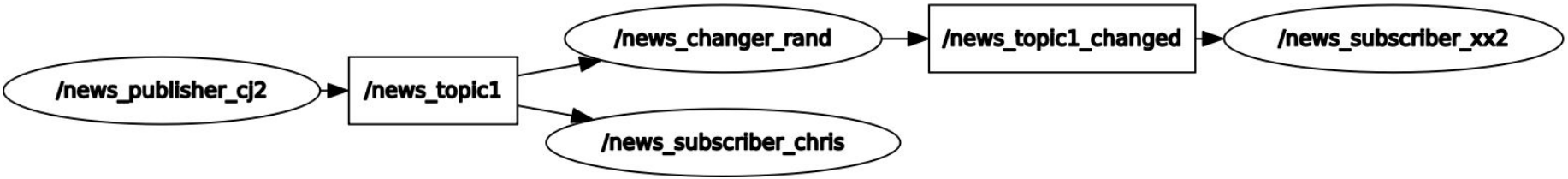
HW2 (1 point)

- Create hw2.launch file (see next slide as an example)
- Create news_publisher_yourName2.cpp - This file reads /news_topic1_pub_freq set with 1.5 and use the value to set Rate object.
- Create news_changer_rand.cpp - this changes the /news_topic1 message by replacing character with ‘X’ at random location of the message string. The modified message is published onto /news_topic1_changed.
- Create news_subscriber_xx2.cpp - this program displays the modified message, frequency parameter, and message counter from 1. See slide #15.
- Due Feb 13, 5:45pm. Submit news_publisher_yourName2.cpp, news_changer_rand.cpp, news_subscriber_xx2.cpp, and hw2.launch files; demonstrate your program on Feb 13 before 6:10pm.

hw2.launch file example

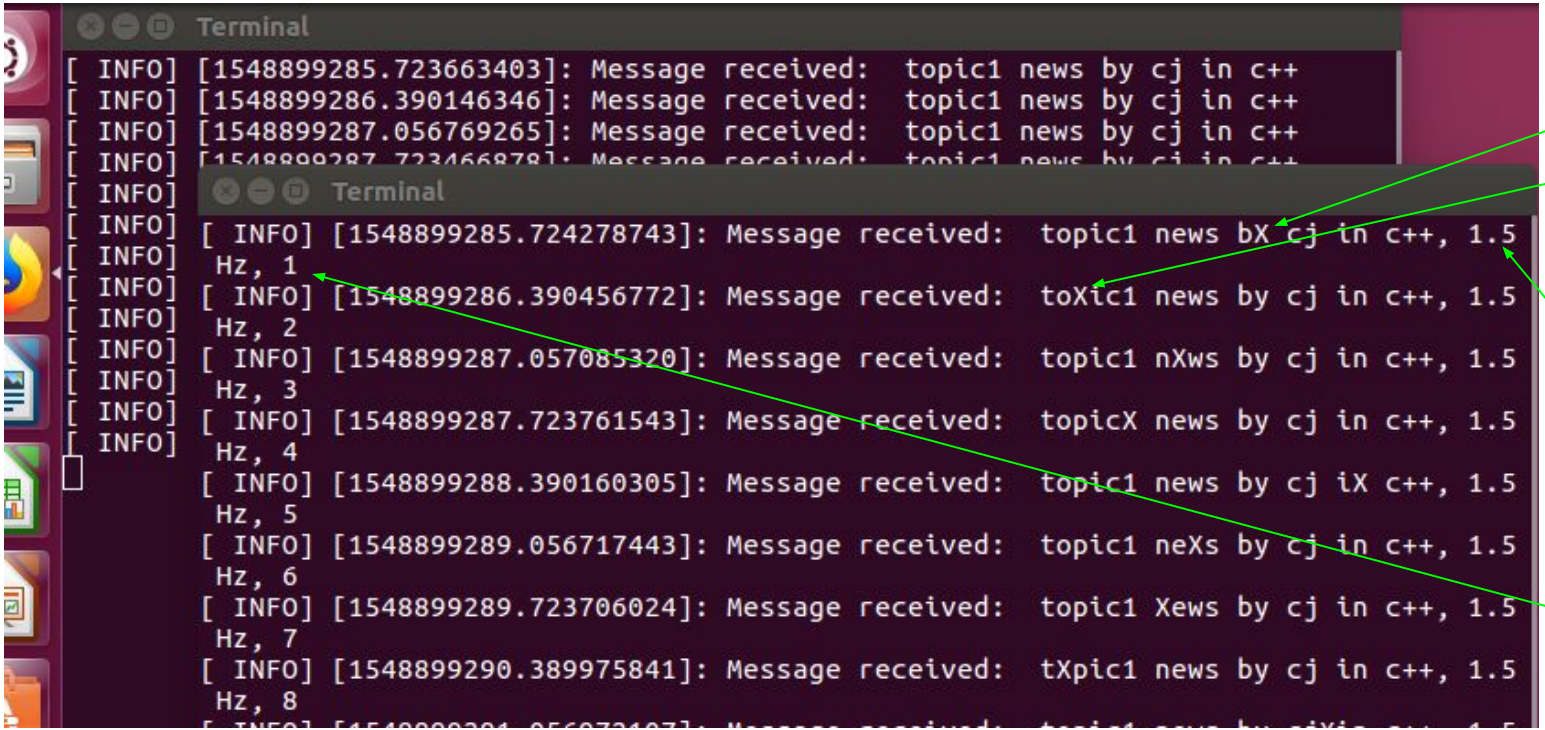
```
<launch>
    <param name="/news_topic1_pub_freq" type="double" value="1.5" />

    <node name="news_publisher_cj2" pkg="my_ros_tutorials" type="news_publisher_cj2" />
    <node name="news_subscriber_chris" pkg="my_ros_tutorials" type="news_subscriber_chris"
          output="screen" launch-prefix="gnome-terminal -e" />
    <node name="news_changer_rand" pkg="my_ros_tutorials" type="news_changer_rand" />
    <node name="news_subscriber_xx2" pkg="my_ros_tutorials" type="news_subscriber_xx2"
          output="screen" launch-prefix="gnome-terminal -e" />
</launch>
```

HW2 rqt_graph



Expected HW2 roslaunch run results

```
cjchung@Robofest:~/catkin_ws/src/my_ros_tutorials/launch$ roslaunch my_ros
_tutorials hw2.launch
```



'X' is replaced at random location

Frequency "parameter" read from ROS Master

Message counter from 1

More about news_changer_rand.cpp


```
topic1 new by cj in c++
```

If random number generated is 3, modified message will be

```
topXc1 new by cj in c++
```

AGITR Reading Assignment -

Chapters 1, 2, 3, 5, 6, 7