

ROS over Multiple Machines

This manual explains the process for connecting ROS Nodes on multiple machines **connected to the same network**.

Example case for 2 Computers on same network:

Hostname	comp1	comp2
IP	192.168.77.11	192.168.77.12

Case 1: Using On-board ROS :-

➔ Configuring ROS master:

There should be only **1 master/ ROS core** for handling the system. In this example, let the **comp1** be the master. Hence, **roscore** is run on this computer.

```
comp1 : $ roscore
```

This should start the **roscore** displaying the *master's IP address* and the *port number*. **11311** is the default port of the master.

➔ Configuring Slaves:

The slave machines need to know the masters's as well as their own IP address.

- **comp1** can act as a slave to itself along with being the master

Specify ROS master's IP:

```
comp1 : $ export ROS_MASTER_URI=http://192.168.77.11:11311
```

Specify slave machine's own IP:

```
comp1 : $ export ROS_IP=192.168.77.11
```

- **comp2** will only act as a slave machine.

Specify ROS master's IP:

```
comp2 : export ROS_MASTER_URI=http://192.168.77.11:11311
```

Specify slave machine's own IP:

```
comp2 : $ export ROS_IP=192.168.77.12
```

Note: In case of more number of machines, one of them can solely act as the master with the **roscore** running on it and all other nodes running on the slaves.

➔ **Resolve master's name to IP (Optional):**

Modify the “/etc/hosts” file in order to resolve master's name to IP in the slave machine.

- In the slave machine, **comp2**, open the file as **Root user**:

```
comp2 : $ sudo nano /etc/hosts
```

- Add the line to the “hosts” file in the format of:
 <Master's IP Address> <Master's Name>
 192.168.77.11 comp1
- Save and close the file.

Note: The commands for specifying the IP address of ROS master and the slave can also be added to a bash file and sourced when required instead of typing it everytime.

Case 2: Using MATLAB-ROS :-

Robotics System Toolbox enables one to interface with ROS and use ROS functionality in MATLAB and Simulink. One can connect to a ROS network, collect data, send and receive one's own messages, and deploy code to a standalone system.

➔ Configuring ROS Master:

To initialize ROS master using MATLAB-ROS, run the following command in the **Matlab Command Window**.

For comp1,

```
>> rosininit
```

This should start the **roscore** and display the *IP address of ROS master* and the *port numbers of ROS Master* and the *global node* on the command window.

Note: For running a ROS node in the master, the Matlab code just needs to be run and no separate configurations are required.

➔ Configuring Slaves:

Before running the slave nodes, MATLAB-ROS must be initialized with a master in the network. To do so, the *IP address* of the ROS master must be passed as an argument to the “rosinit” function.

Eg: rosininit('<Master's IP Address>')

For comp2,

```
>> rosininit('192.168.77.11')
```

Note: In case of one of the computer with MATLAB-ROS and other with On-Board ROS, the nodes can communicate over the network using the corresponding configurations.