

目录

1. Full connectivity
 1. Basic check 1: self ping
 2. Basic check 2: ping between machines
 3. Further check: netcat
2. Name resolution
 1. Setting a name explicitly
 2. Single machine configuration
 3. Configuring `/etc/hosts`
 4. Using `machinename.local`
3. What about firewalls?
4. Debugging network problems
5. Timing issues, TF complaining about extrapolation into the future?

ROS is a distributed computing environment. A running ROS system can comprise dozens, even hundreds of nodes, spread across multiple machines. Depending on how the system is configured, any node may need to communicate with any other node, at any time.

As a result, ROS has certain requirements of the network configuration:

- There must be complete, bi-directional connectivity between all pairs of machines, on all ports.
- Each machine must advertise itself by a name that all other machines can resolve.

In the following sections, we'll assume that you want to run a ROS system on two machines, with the following hostnames and IP addresses:

- `marvin.example.com` : `192.168.1.1`
- `hal.example.com` : `192.168.1.2`

Note that you only need to run one master; see [ROS/Tutorials/MultipleMachines](#) ([/ROS/Tutorials/MultipleMachines](#)).

1. Full connectivity

First of all, **hal** and **marvin** need full bi-directional connectivity, on all ports.

1.1 Basic check 1: self ping

You can check for basic connectivity with `ping`.

Try to ping each machine from itself, i.e. ping **hal** from **hal**:

```
ssh hal
ping hal
```

 *Problem: cannot ping hal: this means that hal is not configured properly.*

See "Name Resolution" section below.

1.2 Basic check 2: ping between machines

Ping **marvin** from **hal**:


```
ssh hal
ping marvin
```

You should see something like:

```
PING marvin.example.com (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=63 time=1.868 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=63 time=2.677 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=63 time=1.659 ms
```

Also try pinging **hal** from **marvin**:

```
ssh marvin
ping hal
```

 **Problem:** cannot ping each other. This means that your machines cannot see each other.

Additional check: try pinging the IP address instead of the hostname. If this does not work, your machines are not on the same network and you will need to reconfigure your network. If the additional check passes, see "Name Resolution" below.

1.3 Further check: netcat

`ping` only checks that ICMP packets can get between the machines, which isn't enough. You need to make sure that you can communicate over all ports. This is difficult to check completely, because you'd have to iterate over approximately 65K ports.

In lieu of a complete check, you can use `netcat` to try communicating over an arbitrarily selected port. Be sure to pick a port greater than 1024; ports below 1024 require superuser privileges. Note that the `netcat` executable may be named `nc` on some distributions.

First try communicating from **hal** to **marvin**. Start `netcat` listening on **marvin**:

```
ssh marvin
netcat -l 1234
```

Then connect from **hal**:

```
ssh hal
netcat marvin 1234
```

If the connection is successful, you will be able to type back and forth between the two consoles, like an old-fashioned chat program.

Now try it the other direction. Start `netcat` listening on **hal**:

```
ssh hal
netcat -l 1234
```

Then connect from **marvin**:

```
ssh marvin
netcat hal 1234
```

2. Name resolution

When a ROS node advertises a topic, it provides a hostname:port combination (a URI) that other nodes will contact when they want to subscribe to that topic. It is important that the hostname that a node provides can be used by all other nodes to contact it. The ROS client libraries use the name that the machine reports to be its hostname. This is the name that is returned by the command `hostname`.

2.1 Setting a name explicitly

If a machine reports a hostname that is not addressable by other machines, then you need to set either the `ROS_IP` or `ROS_HOSTNAME` environment variables (more (/ROS/EnvironmentVariables#ROS_IP.2BAC8-ROS_HOSTNAME)).

2.1.1 Example

Continuing the example of **marvin** and **hal**, say we want to bring in a third machine. The new machine, named **artoo**, uses a DHCP address, say 10.0.0.1, and other machines cannot resolve the hostname **artoo** into an IP address (this should not happen on a properly configured DHCP-managed network, but it is a common problem).


In this situation, neither **marvin** nor **hal** are able to ping **artoo** by name, and so they would not be able to contact nodes that advertise themselves as running on **artoo**. The fix is to set `ROS_IP` in the environment before starting a node on **artoo**:

```
ssh 10.0.0.1 # We can't ssh to artoo by name
export ROS_IP=10.0.0.1 # Correct the fact that artoo's address can't be resolved
<start a node here>
```

A similar problem can occur if a machine's name is resolvable, but the machine doesn't know its own name. Say **artoo** can be properly resolved into 10.0.0.1, but running `hostname` on **artoo** returns **localhost**. Then you should set `ROS_HOSTNAME`:

```
ssh artoo # We can ssh to artoo by name
export ROS_HOSTNAME=10.0.0.1 # Correct the fact that artoo doesn't know its name
<start a node here>
```

2.2 Single machine configuration

If you just want to run tests on your local machine (like to run the  ROS Tutorials (<http://www.ros.org/wiki/ROS/Tutorials>)), set these environment variables:

```
$ export ROS_HOSTNAME=localhost
$ export ROS_MASTER_URI=http://localhost:11311
```

Then `roscore` should initialize correctly.

2.3 Configuring `/etc/hosts`

Another option is to add entries to your `/etc/hosts` file so that the machines can find each other. The hosts file tells each machine how to convert specific names into an IP address.

For more information on the hosts file, please see [this external tutorial](http://www.faqs.org/docs/securing/chap9sec95.html) (<http://www.faqs.org/docs/securing/chap9sec95.html>).

2.4 Using `machinename.local`

Another way to set `ROS_HOSTNAME` is to use `.local` domain

```
$ export ROS_HOSTNAME=ubuntu.local
$ export ROS_MASTER_URI=http://ubuntu.local:11311
```

This is useful when you have a Ubuntu system named “ubuntu” on your network, it can be accessed at the address “ubuntu.local”. To do this, Avahi automatically takes over all DNS requests ending with “.local” and prevents them from resolving normally.

Sometimes, system is unable to resolve to `.local` domain. When you encounter such issue, apart from following diagnostics mentioned above, check whether your avahi service is running.

```
$ systemctl is-active avahi-daemon.service
```

If required, you can restart avahi service as follows:

```
$ systemctl restart avahi-daemon.service
```

3. What about firewalls?

If there is a firewall, or other obstruction, between a pair of machines that you want to use with ROS, you need to create a virtual network to connect them. We recommend [openvpn](http://openvpn.net/) (<http://openvpn.net/>).

4. Debugging network problems

Try `roswtf` (`/roswtf`) and `rqt_graph` (`/rqt_graph`).

Also have a look at the ROS/Troubleshooting (`/ROS/Troubleshooting`) page for more information on common problems.

5. Timing issues, TF complaining about extrapolation into the future?

You may have a discrepancy in system times for various machines. You can check one machine against another using

```
ntptime -q other_computer_ip
```



If there is a discrepancy, install chrony (for Ubuntu, **sudo apt-get install chrony**) and edit the chrony configuration file (**/etc/chrony/chrony.conf**) on one machine to add the other as a server. For instance, on the PR2, computer c2 gets its time from c1 and thus has the following line:

```
server c1 minpoll 0 maxpoll 5 maxdelay .05
```

That machine will then slowly move its time towards the server. If the discrepancy is enormous, you can make it match instantly using

```
/etc/init.d/chrony stop  
ntptime other_computer_ip  
/etc/init.d/chrony start
```

(as root) but large time jumps can cause problems, so this is not recommended unless necessary.

If you are using wifi and are not getting any synchronisation try to set maxdelay higher ( should be bigger than the expected round-trip delay (<http://chrony.tuxfamily.org/manual.html#server>)). For isolated networks look  here (<http://chrony.tuxfamily.org/manual.html#isolated-networks>).

Except where otherwise

noted, the ROS wiki is Wiki: ROS/NetworkSetup (2018-08-08 09:08:07由Rahul Bhadani (/Rahul%20Bhadani)编辑)

licensed under the

Creative Commons Attribution 3.0 (<http://creativecommons.org/licenses/by/3.0/>)

Brought to you by:  Open Source Robotics Foundation

(<http://www.osrfoundation.org>)