



POLYTECHNIC SCHOOL OF THE UNIVERSITY OF NANTES
COMPUTER SCIENCE DEPARTMENT
RESEARCH AND DEVELOPMENT DOCUMENT

Installation Guide : MP-OLSR on Raspberry Pi using Raspbian

Benjamin MOLLÉ & Denis SOURON

February 2015

supervised by Benoît PARREIN

— Équipe IVC —

INSTITUT DE RECHERCHE EN COMMUNICATIONS ET EN
CYBERNÉTIQUE DE NANTES



Warning

All rights reserved. No part of this report may be reproduced, stored, published or transmitted by any means and in any form without prior permission.

Any copy, by xerography, photography, photocopy, film, electronic support, or any other, is an infringement of the copyright.

Installation Guide : MP-OLSR on Raspberry Pi using Raspbian

Benjamin MOLLÉ & Denis SOURON

Abstract

This document is a guide on how to install MP-OLSR on Raspberry Pi. A Raspberry Pi with Raspbian on it is required. Every part of the installation is detailed here. Moreover, starting instructions are available at the end of the document.

Warning

This guide requires superuser's privileges. You may encounter some issues without having it. Please make sure you have superuser's privileges for each part of the document.

Contents

1	Installation	6
1.1	Get an old Linux Kernel	6
1.2	MP-OLSR files	6
1.3	Required librairies and programs	7
1.3.1	Packages from repositories	7
1.3.2	Installation of Bzip2	7
1.3.3	Installation of Boost 1.38	7
1.3.4	Installation of Libnfnctlink	7
1.3.5	Installation of Libnetfilter_queue	8
1.4	Compilation of parts	8
1.4.1	Modification in netfilter_ipv4.h	8
1.4.2	Compilation of ser_module	8
1.4.3	Restoration of netfilter_ipv4.h	8
1.4.4	Link with other libraries	8
1.4.5	Compilation of ser_routing	9
1.4.6	Compilation of ser_valider_dev	9
1.4.7	Compilation of OLSRd	9
1.4.8	Compilation of OLSRd plugin	9
1.5	Installation of MP-OLSRd	9
1.5.1	Installation of OLSRd	9
1.5.2	Installation of ser_routing	9
2	Configuration	10
3	Running	11



Installation

1.1 Get an old Linux Kernel

Before we can install MP-OLSR on a Raspberry Pi, we need to get a Kernel 3.2. So the first step is to update the system and pick up a Kernel 3.2 in the repositories.

```
apt-get update
apt-get upgrade
rpi-update
```

```
apt-get install linux-headers-3.2.0-4-* linux-image-3.2.0-4-rpi
```

Then, the Kernel has to be defined for the start-up. Raspberry Pi do not use GRUB and only boot on the image called `kernel.img` in `/boot` directory. The downloaded kernel image from repositories should be in the same folder. Here are the instructions to update the Kernel.

```
cd /boot
cp kernel.img kernel_backup.img
cp vmlinuz-3.2.0-4-rpi kernel.img
```

```
reboot
```

The new version can be checked via the command `uname -a`. The Raspberry may not boot after this step. In which case the system can be recovered by accessing the `/boot` folder located in a standalone FAT partition of the SD card.

1.2 MP-OLSR files

Copy and paste the file `olsrd.conf` in the folder `/etc`. The files should be available via `/etc/olsrd.conf`

Extract the archive file `mpolsr.zip` in the folder `/opt`. This location is not mandatory at all but it will be the one used in this document. Otherwise, the instructions have to be adapted to the new location.

1.3 Required libraries and programs

1.3.1 Packages from repositories

```
apt-get install python-dev libbz2-dev bison flex automake g++
```

1.3.2 Installation of Bzip2

```
wget http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz
tar xzvf bzip2-1.0.6.tar.gz
cd bzip2-1.0.6
make
make install
```

1.3.3 Installation of Boost 1.38

Warning : This part is time consuming. This step requires at least 6 hours.

```
mkdir -p /opt/boost
cd /opt/boost
wget http://sourceforge.net/projects/boost/files/boost/1.38.0/
    boost_1_38_0.tar.gz
tar xvzf boost_1_38_0.tar.gz
cd boost_1_38_0
./configure
make
make install
echo "/usr/local/lib/" >>/etc/ld.so.conf
ldconfig
```

1.3.4 Installation of Libnfnetlink

```
mkdir -p /opt/libnfnetlink
cd /opt/libnfnetlink
wget http://ftp.netfilter.org/pub/libnfnetlink/libnfnetlink-0.0.41.tar
    .bz2
tar xvjf libnfnetlink-0.0.41.tar.bz2
cd libnfnetlink-0.0.41
./configure
make
make install
```

1.3.5 Installation of Libnetfilter_queue

```
mkdir -p /opt/libnetfilter_queue
cd /opt/libnetfilter_queue
wget http://ftp.netfilter.org/pub/libnetfilter_queue/
    libnetfilter_queue-0.0.17.tar.bz2
tar xvjf libnetfilter_queue-0.0.17.tar.bz2
cd libnetfilter_queue-0.0.17
PKG_CONFIG_PATH=/usr/local/lib/pkgconfig ./configure
make
make install
```

1.4 Compilation of parts

1.4.1 Modification in netfilter_ipv4.h

Modify the file `"/usr/include/linux/netfilter_ipv4.h"`

At line 13, replace `: #include <limits.h>` by `#include <linux/limits.h>`.

1.4.2 Compilation of ser_module

```
cd /opt/mpolsr/mpolsr_testbed/src_iptables/ser_module
make
```

1.4.3 Restoration of netfilter_ipv4.h

Modify the file `"/usr/include/linux/netfilter_ipv4.h"`.

At line 13, replace `: #include <linux/limits.h>` by `#include <limits.h>`.

1.4.4 Link with other libraries

For the next compilation, links to an old compatible version are required. The links for this step depend on the version of GCC used in your configuration. The following lines is for GCC 4.6. Please check the files in `/usr/local/lib`, then follow and adapt the instructions below.

```
cd /usr/local/lib/
ln -s libboost_date_time-gcc46-mt.so libboost_date_time-gcc41-mt.so
ln -s libboost_serialization-gcc46-mt.so \
    libboost_serialization-gcc41-mt.so
ln -s libboost_system-gcc46-mt.so libboost_system-gcc41-mt.so
ln -s libboost_thread-gcc46-mt.so libboost_thread-gcc41-mt.so
```


1.4.5 Compilation of ser_routing

```
cd /opt/mpolsr/mpolsr_testbed/src_iptables/ser_routing
./configure
make
make install
```

1.4.6 Compilation of ser_valider_dev

```
export LD_LIBRARY_PATH=/usr/local/lib/:\$LD_LIBRARY_PATH
cd /opt/mpolsr/mpolsr_testbed/src_iptables/ser_valider_dev/
make
```

1.4.7 Compilation of OLSRd

```
cd /opt/mpolsr/mpolsr_testbed/src_olsr/olsrd-0.5.6-r2/
chmod a+x gcc-warnings ld-warnings
make
make libs
```

1.4.8 Compilation of OLSRd plugin

```
cd /opt/mpolsr/mpolsr_testbed/src_olsr/olsrd-0.5.6-r2/lib/sereadmo
make clean
ln -s /usr/local/include/boost-1_38/boost/ /usr/local/include/boost
./install_plugin.sh
```

1.5 Installation of MP-OLSRd

1.5.1 Installation of OLSRd

```
cd /opt/mpolsr/mpolsr_testbed/src_olsr/olsrd-0.5.6-r2/
make install_all
```

Warning : The file /etc/olsrd.conf must not be erased.

1.5.2 Installation of ser_routing

```
cd /opt/mpolsr/mpolsr_testbed/src_iptables/ser_routing
chmod a+x configure
./configure
make
```



Configuration

The configuration of MP-OLSR is quite easy. The interfaces have to be set in the file `/etc/olsrd.conf`. In the provided file MP-OLSR is configured for using only the interface `eth0`. Check OLSRd configuration online for further information.¹

1. <http://www.olsr.org/docs/olsrd.conf.5.html>



Running

In a first terminal :

```
cd /opt/mpolsr/mpolsr_testbed/src_iptables/ser_module
./inst_module.csh
olsrd -nofork
```

The script `inst_module.csh` adds an emulated device named `sereadm0` located in `/dev/sereadm0`. The `-nofork` option allows to have a foreground OLSRd in which good working of OLSRd and MP-OLSR can be confirmed.

Then in a second terminal :

```
cd /opt/mpolsr/mpolsr_testbed/src_iptables/ser_routing/src
./ser_routing
```

This server is mandatory for MP-OLSR. Links and nodes states are periodically printed in this terminal, such as new links, removed nodes, etc. . . .