OLSR: installation and configuration for Node Position Update

OLSR Version: Olsrd 0.6.5
OS : ubuntu 12.04 and ubuntu 10.04

Steps:

1. Download OLSR (with pud plugin , version 0.6.3 or later). In the latest version 0.6.5 , the PUD plug-in can be used to dynamically update the position information of the nodes. So, I am using this version from the git.(`git clone git://olsr.org/olsrd.git`)
2. To install olsrd, just go to the olsrd directory and run the commands below:
```
sudo make
sudo make install
```
3. I have installed few plug-ins as well. Go to olsrd/lib directory, then you can see the plug-ins that can be installed. I have installed httpinfo (for accessing the olsr through browser) , txtinfo (for getting olsr network information like topology, MPR, etc) along with pud.
   To install the plug-ins we you just need to make and install as shown above. After installing you can configure the plug-ins details, e.g. port number, in the file /etc/olsrd.conf
4. Configuring httpinfo plug-in:
```
LoadPlugin "olsrd_httpinfo.so.0.1"

{

 PlParam "port" "8080"

  PlParam  "Net" "0.0.0.0 0.0.0.0"

}
```
5. Configuring txtpinfo plug-in:
```
LoadPlugin "olsrd_txtinfo.so.0.1"

{

   # port number the txtinfo plugin will be listening, default 2006

   PlParam     "Accept"   "127.0.0.1"

}
```
6. Configuring pud plug-in:(Special thanks to Ferry Huberts and Teco Boot)
   - Enable Smart gateway :
```
 SmartGateway yes
```

- *LoadPlugin "/usr/local/lib/olsrd_pud.so.1.1.0"*

```
{
# only some of the parameters are shown here
# please refer to the appendix to access the  whole olsrd.conf file
.
.
PlParam     "rxNonOlsrIf"              "wlan1"
PlParam "positionFile"                "/home/bidur/Desktop/position"
PlParam "positionFilePeriod" "1000"
PlParam     "txNonOlsrIf"              "wlan1"
PlParam     "uplinkAddr"               "127.0.0.1"
.
}
```

7. Run olsrd:
   sudo olsrd

8. Test httpinfo and pud:
   Go to browser and type: *127.0.0.1:8080*
   You should be able to see the olsr web interface.
   You should also be able to see Position tab if your pud configuration is done properly.

9. Test the txtinfo:
   Go to browser and type: *127.0.0.1:2006*
   You should be able to see olsr links, topology, neighbors, etc.
   - If you type *127.0.0.1:2006/topo* then you will just see topology only.

   Try to replace /*topo* by */all, /neigh , /links, /routes* and see the difference.
   - You can even use telnet to access these data. Just go to the terminal and type:*telnet*
     127.0.0.1 2006
     /all

10. Test pud plugin:
    - read the GPS feed and update the positionFile on the fly. A sample positionFile is given in the appendix. Changing the contents in the positionFile will be reflected within the pud plugin interface.
    - Tcpdump can be run to access the location of other olsr nodes.

    ```
    tcpdump -nlAs 0 -i $lan udp port 2240 | grep NBSX
    ```

    The tcpdump shows the position details of the olsr nodes in NMEA like NBSX statements. For example:

```
E..y..@.............\...e.F$PNBSX,1,0,192.168.8.3,4,192.168.8.3,031212,061359,5,10.00001,N,10.00011,
E,10,10,0,10.00*49
```

I got this tcpdump output in node *192.168.8.6*, this statement gives information of the node *192.168.8.3.* The positin of the node is given as lat/lon/elevation as *10.00001,N/10.00011,E/10.* For more detail information please refer to the pud documentation file (*olsrd/lib/pud/doc/pud.odt)*

APPENDIX

1. positionFile:

```
#########################
#start position file
lat = 10
lon = 10
elv = 10


#end position file
######################
```

NOTE: In the olsr position tab the lat/lon values are expressed in degrees but the NMEA format does not use degree as the unit. To convert the NMEA values to degrees we follow as shown in the example below:
Input(NMEA value): 3120.234 = **31**<u>20.234</u>
Conversion: 31 +(20234/60) =31.3372
Result(Degree) : 31.3372

2. olsrd.conf

```
#
# OLSR.org routing daemon config file
# This file contains the usual options for an ETX based
# stationary network without fisheye
# (for other options see olsrd.conf.default.full)
#
# Lines starting with a # are discarded
#

#### ATTENTION for IPv6 users ####
# Because of limitations in the parser IPv6 addresses must NOT
# begin with a ":", so please add a "0" as a prefix.

###########################
### Basic configuration ###
###########################
# keep this settings at the beginning of your first configuration file

# Debug level (0-9)
# If set to 0 the daemon runs in the background, unless "NoFork" is set to true
# (Default is 1)

DebugLevel  2

# IP version to use (4 or 6)
# (Default is 4)

# IpVersion 4

#################################
### OLSRd agent configuration ###
#################################
# this parameters control the settings of the routing agent which are not
# related to the OLSR protocol and it's extensions

# FIBMetric controls the metric value of the host-routes OLSRd sets.
# - "flat" means that the metric value is always 2. This is the preferred value
#   because it helps the linux kernel routing to clean up older routes
# - "correct" use the hopcount as the metric value.
# - "approx" use the hopcount as the metric value too, but does only update the
#   hopcount if the nexthop changes too
# (Default is "flat")

# FIBMetric "flat"

#######################################
### Linux specific OLSRd extensions ###
#######################################
# these parameters are only working on linux at the moment, but might become
# useful on BSD in the future

# SrcIpRoutes tells OLSRd to set the Src flag of host routes to the originator-ip
# of the node. In addition to this an additional localhost device is created
# to make sure the returning traffic can be received.
# (Default is "no")

# SrcIpRoutes no

# Specify the proto tag to be used for routes olsr inserts into kernel
# currently only implemented for linux
# valid values under linux are 1 .. 254
# 1 gets remapped by olsrd to 0 UNSPECIFIED (1 is reserved for ICMP redirects)
# 2 KERNEL routes (not very wise to use)
# 3 BOOT (should in fact not be used by routing daemons)
# 4 STATIC
# 8 .. 15 various routing daemons (gated, zebra, bird, & co)
# (defaults to 0 which gets replaced by an OS-specific default value
# under linux 3 (BOOT) (for backward compatibility)

# RtProto 0

# Activates (in IPv6 mode) the automatic use of NIIT
# (see README-Olsr-Extensions)
# (default is "yes")

# UseNiit yes

# Activates the smartgateway ipip tunnel feature.
# See README-Olsr-Extensions for a description of smartgateways.
# (default is "no")

 SmartGateway yes

# Allows the selection of a smartgateway with NAT (only for IPv4)
# (default is "yes")

# SmartGatewayAllowNAT yes

# Defines what kind of Uplink this node will publish as a
# smartgateway. The existence of the uplink is detected by
# a route to 0.0.0.0/0, ::ffff:0:0/96 and/or 2000::/3.
# possible values are "none", "ipv4", "ipv6", "both"
# (default is "both")

# SmartGatewayUplink "both"

# Specifies if the local ipv4 uplink use NAT
# (default is "yes")

 #SmartGatewayUplinkNAT yes

# Specifies the speed of the uplink in kilobit/s.
# First parameter is upstream, second parameter is downstream
# (default is 128/1024)

 SmartGatewaySpeed 128 1024

# Specifies the EXTERNAL ipv6 prefix of the uplink. A prefix
# length of more than 64 is not allowed.
# (default is 0::/0

# SmartGatewayPrefix 0::/0

################################
### OLSR protocol settings ###
################################

# HNA (Host network association) allows the OLSR to announce
# additional IPs or IP subnets to the net that are reachable
# through this node.
# Syntax for HNA4 is "network-address    network-mask"
# Syntax for HNA6 is "network-address    prefix-length"
# (default is no HNA)
Hna4
{
# Internet gateway
# 0.0.0.0   0.0.0.0
# specific small networks reachable through this node
# 15.15.0.0 255.255.255.0
}
Hna6
{
# Internet gateway
#   0::                     0
# specific small networks reachable through this node
#   fec0:2200:106:0:0:0:0:0 48
}

##################################
### OLSR protocol extensions ###
##################################

# Link quality algorithm (only for lq level 2)
# (see README-Olsr-Extensions)
# - "etx_float", a floating point  ETX with exponential aging
# - "etx_fpm", same as ext_float, but with integer arithmetic
# - "etx_ff" (ETX freifunk), an etx variant which use all OLSR
#   traffic (instead of only hellos) for ETX calculation
# - "etx_ffeth", an incompatible variant of etx_ff that allows
#   ethernet links with ETX 0.1.
# (defaults to "etx_ff")

#LinkQualityAlgorithm    "etx_ff"
```

```
# Fisheye mechanism for TCs (0 meansoff, 1 means on)
# (default is 1)

LinkQualityFishEye  0

#######################################
### Example plugin configurations ###
#######################################
# Olsrd plugins to load
# This must be the absolute path to the file
# or the loader will use the following scheme:
# - Try the paths in the LD_LIBRARY_PATH
#   environment variable.
# - The list of libraries cached in /etc/ld.so.cache
# - /lib, followed by /usr/lib
#
# the examples in this list are for linux, so check if the plugin is
# available if you use windows/BSD.
# each plugin should have a README file in it's lib subfolder

# LoadPlugin "olsrd_txtinfo.dll"

#LoadPlugin "olsrd_mprfrontback.so.0.1"
#{
#   PlParam     "port" "102"
#}


#LoadPlugin "olsrd_gpsperiodic.so.0.1"
#{
#   PlParam     "port" "103"
#}


#LoadPlugin "olsrd_locinfo.so.0.1"
#{
#   PlParam     "port" "101"
#}

LoadPlugin "olsrd_txtinfo.so.0.1"
{
   # port number the txtinfo plugin will be listening, default 2006
   #PlParam     "port"   "81"
    # ip address that can access the plugin, use "0.0.0.0"
    # to allow everyone
```

```
  PlParam     "Accept"   "127.0.0.1"
}

#LoadPlugin "olsrd_httpinfo.so.0.1"
LoadPlugin "olsrd_httpinfo.so.0.1"
{
   PlParam "port" "8080"
   PlParam  "Net" "0.0.0.0 0.0.0.0"
}

#/usr/local/lib/olsrd_jsoninfo.so.0.0
LoadPlugin "olsrd_jsoninfo.so.0.0"
{
    # the default port is 9090 but you can change it like this:
    #PlParam     "port"   "8080"

    # You can set a "accept" single address to allow to connect to
    # jsoninfo. If no address is specified, then localhost (127.0.0.1)
    # is allowed by default.  jsoninfo will only use the first "accept"
    # parameter specified and will ignore the rest.

    # to allow a specific host:
    #PlParam      "accept" "172.29.44.23"
    # if you set it to 0.0.0.0, it will accept all connections
    PlParam      "accept" "0.0.0.0"

    # specify a UUID for this node to track it for debugging
    #PlParam      "UUIDFile" "/etc/olsrd/olsrd.uuid"
}




#LoadPlugin "olsrd_smon.so.0.2"
#{
#       PlParam "port" "33333"
#         PlParam     "Accept"   "127.0.0.1"
#}

LoadPlugin "/usr/local/lib/olsrd_pud.so.1.1.0"
{
    # nodeIdType is used to indicate the type of the nodeId field and is a
    #           number in the range 0-255, with the following meaning:
    #
```

```
#       0 : MAC address of sending interface
#           (nodeId is not relevant)
#       1 : an MSISDN number with 15 digits
#       2 : a Tetra number with 17 digits
#       3 : a DNS name
#       4 : IPv4 address (OLSR main address) of the sending node
#           (nodeId is not relevant)
#       6 : IPv6 address (OLSR main address) of the sending node
#           (nodeId is not relevant)
#       7 : an AIS MMSI number with 9 digits
#       8 : a URN number with 8 digits
#     192 : a 7 digit number conforming to 'Nationaal Nummerplan
#           Brandweer Nederland'
#     193 : a 6 digit number conforming to 'Nationaal Nummerplan
#           Ambulancezorg Nederland'
#     194 : a 4 digit number in the range [1, 8191]
#
#     other numbers are reserved
#
# Note: setting 4 or 6 is the same for olsrd: both mean 'use the main
#       IP address, which is either an IPv4 or an IPv6 address, depending
#       on the IP version under which olsrd is run.
#
# Default: 4 or 6, depending on the IP version olsrd is using
#
#PlParam     "nodeIdType"                 "4"

# nodeId is the node identification with the meaning as indicated by the
#        nodeIdType parameter. When not set AND the nodeIdType is 3 (DNS
#        name) then the hostname is used.
#
# Note: Both the nodeIdType and nodeId fields are transported over
#       OLSR, so care must be taken to keep the size of the nodeId
#       parameter down.
#
# Default: none
#
#PlParam     "nodeId"                     "host.example.com"


#
# RX Parameters
#

# rxNonOlsrIf is a network interface on the host on which the plugin will
#           listen for GPS multicasts. Multiple such interfaces can be
#           specified by specifying the parameter multiple times.
#
# Default: none
#
PlParam     "rxNonOlsrIf"               "wlan1"

# rxAllowedSourceIpAddress is an IP address from which the plugin is
#                       allowed to process/parse GPS sentences. When
#                       this parameter is not configured then GPS
#                       sentences from ALL IP addresses are processed.
#                       Multiple IP addresses can be specified by
#                       specifying the parameter multiple times.
#
# Default: none
#
# PlParam     "rxAllowedSourceIpAddress"     "127.0.0.1"

# rxMcAddr is the multicast address on which the plugin will listen for GPS
#         multicasts.
#
# Default: 224.0.0.224 (IPv4) or FF02:0:0:0:0:0:0:1 (IPv6)
#
# PlParam     "rxMcAddr"                  "192.168.8.255"

# rxMcPort is the UDP port on which the plugin will listen for GPS
#         multicasts.
#
# Default: 2240
#
#PlParam     "rxMcPort"                  "2240"

# positionFile is the file that contains the position information that the
#            plugin should use. When this parameter is set then the
#            file is read during olsrd startup. It has no default, but an
#            example file is available:
#              /usr/share/doc/olsrd/olsrd.pud.position.conf
 PlParam "positionFile"                  "/home/bidur/Desktop/position"

# Specifies the period in milliseconds on which to read the positionFile
# (if it changed) and activate its new setting for the position.
# This setting is only relevant if positionFile has been configured.
# A setting of zero disables dynamic updates, the positionFile is then only
# read during olsrd startup.
#
```

```
# Default: 0
#
PlParam "positionFilePeriod" "1000"

#
# TX Parameters
#

# txNonOlsrIf is a network interface on the host on which the plugin will
#           transmit GPS multicasts that were received through the OLSR
#           network. Multiple such interfaces can be specified by
#           specifying the parameter multiple times.
#
# Default: none
#
PlParam     "txNonOlsrIf"                "wlan1"

# txMcAddr is the multicast address on which the plugin will transmit GPS
#         multicasts that were received through the OLSR network.
#
# Default: 224.0.0.224 (IPv4) or FF02:0:0:0:0:0:0:1 (IPv6)
#
#PlParam     "txMcAddr"                  "224.0.0.224"

# txMcPort is the UDP port on which the plugin will transmit GPS multicasts
#         that were received through the OLSR network.
#
# Default: 2240
#
#PlParam     "txMcPort"                  "2240"

# txTtl is the TTL that is used when transmitting GPS multicasts that were
#       received through the OLSR network
#
# Default: 1
#
#PlParam     "txTtl"                     "1"

# txNmeaMessagePrefix is the NMEA message prefix of GPS multicasts that the
#                 plugin transmits. It must be exactly 4 characters
#                 long.
#
# Default: NBSX
#
#PlParam     "txNmeaMessagePrefix"         "NBSX"
```

```
#
# Uplink Parameters
#

# uplinkAddr is the IP address to which the plugin will transmit GPS
#           position updates. When not set, no uplink messages will be
#           sent.
#
# Default: none
#
PlParam     "uplinkAddr"                  "127.0.0.1"

# uplinkPort is the UDP port to which the plugin will transmit GPS position
#           updates. Can't be the same as the downlink port.
#
# Default: 2241
#
PlParam     "uplinkPort"                  "2241"

# downlinkPort is the UDP port on which the plugin will receive GPS position
#             updates. Can't be the same as the uplink port.
#             The downlink is only active when a proper uplink has been
#             configured.
#
# Default: 2242
#
PlParam     "downlinkPort"                "2242"


#
# OLSR Parameters
#

# olsrTtl is the TTL that is used when sending messages over the OLSR
#         networks
#
# Default: 64
#
#PlParam     "olsrTtl"                    "64"


#
# Update Parameters
```

```
#

# updateIntervalStationary is the interval (in seconds) between position
#                        updates sent over the OLSR network when the
#                        node is stationary
#
# Default: 60
#
PlParam     "updateIntervalStationary"     "5"

# updateIntervalMoving is the interval (in seconds) between position
#                    updates sent over the OLSR network when the
#                    node is moving
#
# Default: 5
#
PlParam     "updateIntervalMoving"         "5"

# uplinkUpdateIntervalStationary is the interval (in seconds) between
#                              position updates sent over the uplink when
#                              the node is stationary
#
# Default: 180
#
PlParam     "uplinkUpdateIntervalStationary"  "5"

# uplinkUpdateIntervalMoving is the interval (in seconds) between position
#                          updates sent over the OLSR network when the
#                          node is moving
#
# Default: 15
#
PlParam     "uplinkUpdateIntervalMoving"      "5"

# gatewayDeterminationInterval is the interval (in seconds) on which
#                            determination of the best gateway is
#                            performed
#
# Default: 1
#
#PlParam     "gatewayDeterminationInterval"    "1"

# movingSpeedThreshold is the speed from which we consider the node is
#                    moving
#
```

```
# Default: 9
#
PlParam     "movingSpeedThreshold"        "1"

# movingDistanceThreshold is the distance from the previous position from
#                         which we consider the node is moving
#
# Default: 50
#
PlParam     "movingDistanceThreshold"     "1"

# dopMultiplier One of the situations that is seen as movement is when the
#               current position with its uncertainty circle no longer
#               overlaps the last transmitted position with its uncertainty
#               circle. This parameter is used to adjust the sizes of these
#               uncertainty circles: setting it to a value less than 1.0
#               will make both uncertainty circles smaller by this factor,
#               resulting in earlier movement detection. Setting it to a
#               value larger than 1.0 will detect movement later.
#
# Default: 2.5
#
#PlParam     "dopMultiplier"              "2.5"

# defaultHdop is the default value that is taken for HDOP (in meters) in
#             determining whether we are moving when there is a position
#             available but no HDOP.
#
# Default: 50
#
#PlParam     "defaultHdop"                "50"

# defaultVdop is the default value that is taken for VDOP (in meters) in
#             determining whether we are moving when there is a position
#             available but no VDOP.
#
# Default: 50
#
#PlParam     "defaultVdop"                "50"

# averageDepth is the depth of the position average list, or the number
#              of positions that are averaged to obtain the average
#              position
#
# Default: 5
```

```
#
#PlParam     "averageDepth"              "5"

# hysteresisCountToStationary is the number of position updates that
#                         effectuate a state transition from moving to
#                         stationary that must be received before the
#                         actual transition is taken
#
# Default: 17
#
#PlParam     "hysteresisCountToStationary"  "17"

# hysteresisCountToMoving is the number of position updates that effectuate
#                      a state transition from stationary to moving that
#                      must be received before the actual transition is
#                      taken
#
# Default: 5
#
#PlParam     "hysteresisCountToMoving"      "5"

# gatewayHysteresisCountToStationary is the number of times the gateway
#                         must be the same that effectuate a state transition from
#                         moving to stationary that must be received
#                         before the actual transition is taken
#
# Default: 17
#
#PlParam     "gatewayHysteresisCountToStationary"  "17"

# gatewayHysteresisCountToMoving is the number of gateway updates that
#                      effectuate a state transition from stationary to
#                      moving that must be received before the actual
#                      transition is taken
#
# Default: 5
#
#PlParam     "gatewayHysteresisCountToMoving"      "5"


#
# Other Plugin Parameters
#

# useDeDup determines whether duplicate message detection is to be
```

```
    #       performed. When 0 then no such detection is performed, when 1
    #       then the detection is performed
    #
    # Default: true
    #
    #PlParam     "useDeDup"              "true"

    # deDupDepth the number of messages that are tracked to detect duplucates
    #        messages received from the OLSR network
    #
    # Default: 256
    #
    #PlParam     "deDupDepth"            "256"

    # useLoopback determines whether the message that is sent over the OLSR
    #         network should be immediately looped back, thus pretending
    #         that the message (that is sent by this node) is received from
    #         the OLSR network. When 0 then no loopback is performed, when
    #         1 then the loopback is performed
    #
    # Default: false
    #
    PlParam     "useLoopback"           "true"
}
#################################################
### OLSRD default interface configuration ###
#################################################
# the default interface section can have the same values as the following
# interface configuration. It will allow you so set common options for all
# interfaces.

InterfaceDefaults {
    # Ip4Broadcast      255.255.255.255
}

#########################################
### OLSRd Interfaces configuration ###
#########################################
# multiple interfaces can be specified for a single configuration block
# multiple configuration blocks can be specified

# WARNING, don't forget to insert your interface names here !
Interface "wlan1"
{
    # Interface Mode is used to prevent unnecessary
    # packet forwarding on switched ethernet interfaces
    # valid Modes are "mesh" and "ether"
    # (default is "mesh")

    # Mode "mesh"
    HelloInterval      2.0
#   HelloValidityTime  3.0
    HelloValidityTime  6.0
    TcInterval         5.0
#   TcValidityTime     8.0
    TcValidityTime     15.0
    MidInterval        5.0
#   MidValidityTime    30.0
    MidValidityTime    15.0
    HnaInterval        5.0
#   HnaValidityTime    30.0
    HnaValidityTime    15.0
}
```