

NAME

mmrp – Mobile Mesh Router

SYNOPSIS

mmrp [-hvz] [-d file] [-f file]

DESCRIPTION

mmrp speaks the Mobile Mesh Routing Protocol on one or more IP interfaces. It computes least cost unicast paths to all other nodes within the Mobile Mesh cloud and modifies the IP routing table accordingly. Interfaces and external route advertisements may be specified in a config file or dynamically.

mmrp learns of link events by connecting to the well known Unix socket associated with each interface. A link discovery program like **mmdiscover** must be running on an interface prior to informing **mmrp** to use the interface.

To enhance scalability **mmrp** performs a technique called fish-eye routing where the resolution of a node's map of the network is a function of distance. This reduces the overhead associated with the flooding of LSP's.

OPTIONS

mmrp recognizes the following command line switches:

- d file** Use flags 'file' to enable debug output.
- f file** Use config 'file' instead of */etc/mobilemesh/mmrp.conf*. See below for a description of the config file format.
- h** Display a help message.
- v** Display version information.
- z** Run in the foreground instead of running as a daemon.

Config File Format

By default, **mmrp** reads the file */etc/mobilemesh/mmrp.conf* for configuration parameters. The format of the config file requires each configuration parameter to be on its own line. Comments may be placed in the file by beginning a line with a #. The following configuration parameters are recognized:

- port** <Udp Port Number>
The well known UDP port number that all nodes in the mobile network use to exchange Mobile Mesh Routing Protocol messages.
- alpha** <float value>
where (1.0 <= value <= 2.0). alpha controls the amount of delay imposed before forwarding a neighbor's LSP. The delay is: $\text{Max}(\text{minTxDelay}, \text{Min}(\text{maxTxDelay}, \alpha^{\text{hops}}))$ where hops is the number of hops the LSP has travelled from its source.
- minTxDelay** <unsigned int value>
the minimum delay in seconds that will be imposed before forwarding a neighbor's LSP.
- maxTxDelay** <unsigned int value>
the maximum delay in seconds that will be imposed before forwarding a neighbor's LSP.
- updatePeriod** <unsigned int value>
the number of seconds between sending our own LSP.
- maxAge** <unsigned int value>
the maximum number of seconds that an LSP can live within the network before it is discarded.
- interface** <dev name>
where <dev name> is the name of an interface, eg. eth0, that will participate in the Mobile Mesh Routing Protocol. Multiple interfaces may be specified, each requires its own line.

external <ip address> <netmask> <metric>

where <ip address> and <netmask> specify a range of addresses that this node can reach which are external to the Mobile Mesh cloud. <metric> is the associated cost to reach these addresses from this node.

Unix Sockets

mmp makes extensive use of Unix sockets to supply information to other programs and receive commands. Each well known socket name is appended with the UDP port number which allows multiple copies of **mmp** to execute simultaneously. The list below summarizes each Unix socket:

mmp-c<port>

Configuration socket. Allows clients to dynamically add and delete IP interfaces without restarting the router. This is useful in situations where a system's set of IP interfaces can change.

mmp-i<port>

Import socket. Allows clients to dynamically add and delete external route advertisements. External route advertisements are included as part of the router's Link State Packet and provide a mechanism for a node to advertise a route to an address space that exists outside of the Mobile Mesh cloud.

mmp-t<port>

Topology socket. Allows clients to receive all LSP's stored in the router's database. This mechanism is used by the visualization tool **mmpviz** to learn the network topology.

SEE ALSO

mmdiscover(8), *mmborder(8)*, *mmpviz(8)*

AUTHOR

This program and documentation was written by Kevin H. Grace of The MITRE Corporation <kgrace@mitre.org>. Credit is also given to Mike Butler who developed some of the support classes and lent his keen insight.