B.A.T.M.A.N Daemon HowTo

B.A.T.M.A.N stands for " better approach to mobile ad-hoc networking ", this is a new routing protocol for multi-hop ad-hoc mesh networks. Go to http://open-mesh.net to see more information.
The following document will explain how to install and use the batman daemon.

I. Installing from source

Pre-requirements

- Compile environment and libraries.

1. gcc
2. libc6-dev
3. build-essential
4. binutils
5. makedev
6. make
7. libpthread

- Download the batman daemon code from the website

```
http://open-mesh.net/batman/downloads
```

Compiling

All you have to do is untar and make, then you will see the executable file called "batmand".

```
$ wget http://downloads.open-mesh.net/batman/stable/sources/batmand_0.2-current_sources.tgz
$ tar xzvf batmand_0.2-current_sources.tgz
$ cd batmand_0.2-rv451_sources
$ make
```

If you want reduce the size of executable file, just strip it by executing:

```
$ strip batmand
```

Note that if you want to help us finding a bug in the daemon, please don't strip it.

Installing

Copy “batmand“ to a location somewhere in your path, for example

```
$ cp batmand /usr/sbin/
```

Or start it right from the directory where you compiled it

```
$ ./batmand
```

II. Usage

If you execute batmand -h or -H, you will see the help page, in the following we will explain all parameters and how to work with them in a mesh network.

SYNOPSIS

```
batmand [options] interface [interface interface]
```

DESCRIPTION

You can start batmand without specifying options, but you have to choose at least one interface batman runs on.

```
$ batmand eth1
```

If you have more interfaces, then you just add them behind the first.

```
$ batmand eth1 eth2 eth3
```

The batman daemon can also run on alias interfaces.
Note that we use alias interfaces to separate batman routing protocol and olsr routing protocol.

```
$ batmand eth1:test1 eth2:test2 eth3:test3
```

Note that the batman daemon will take the ip address and the broadcast address from the given interfaces.
Note also that you have to check whether your essid, channel or wifi mode is correct or not.

-o originator interval in ms

Originator means a node transmits broadcast messages (we call them originator message or OGM) to inform the neighboring nodes about its existence.
Originator interval is the time to wait after sending one message and before the batman daemon sends the next message.
The default value is 1000 ms (1 second).
In a mobile network, you may want to detect network changes very quickly, so you need to send message very often, for example, use a value of 500 ms.
In a static network, you can save bandwidth by using a higher value.

```
$ batmand -o 2000 eth1
```

In this case, batmand will wait 2 second until sending the next OGMs.

-d debug level

The debug level can be set to five values.

default:	0 -> debug disabled
allowed values:	1 -> list neighbors
	2 -> list gateways
	3 -> observe batman
	4 -> observe batman (very verbose)

Level 1 just lists the neighbors in your batman network.

The output looks like this

```
Originator        Router (#/128):   Potential routers... [B.A.T.M.A.N. 0.2, MainIF/IP:
eth2 105.131.131.175, UT: 0d 0h 3m]
105.131.83.2      105.131.1.3 (  71):    105.131.1.3 (  71)
105.131.1.2       105.131.1.2 (  52):    105.131.1.2 (  52)
105.131.56.10     105.131.1.4 (  25):   105.131.1.4 (  25),105.131.1.6 ( 15), 105.131.1.7 ( 10),
....
105.131.131.70   105.131.131.70 (121):  105.131.131.70 (121)
```

- In the first line, we will see the version of the batman daemon, main interface, main IP, and uptime.
- In the first column, we can see those IPs which we can reach.
- In the second column, we can see those IPs which we sent our packets to when we want to reach the IP of the first column. The number in the parenthesis indicates the link quality of the connection and the #/128 shows the maximum number of packets.

- In the third column, we can see those IPs which are one hop neighbors and rebroadcasted packets from the originator. The batman daemon will choose the router with the best link quality from the potential router list.

In this case, 105.131.1.2 is a one hop neighbor of 105.131.131.175, because the 105.131.1.2 is originator, router and potential router at the same time. If 105.131.131.175 wants to exchange data with the 105.131.83.2, then it will sent its packets to the 105.131.1.3, because it is the router for this destination.

Level 2 just lists gateways in the batman network.

The output looks like this

Gateway	***Router (#/128)***
105.131.83.5	***105.131.41.1 (57), gw_class 11 - >6 MBit, reliability: 0***
105.131.41.5	***105.131.41.1 (53), gw_class 11 - >6 MBit, reliability: 0***

- In the first column, we can see those IPs which are our gateways.
- In the second column, we can see those IPs which we sent our packets to when we want to reach the IP of the first column. The number in the parenthesis indicates the link quality of the connection and the #/128 shows the maximum number of packets. The "gw_class" means gateway class of the gateway and "11 ->6 MBit" means how much bandwidth the gateway owner wants to share. The reliability means how good the quality of the internet connection is. In this case, "0" means this is the best quality. The reliability number will increase if the quality is poor.

Level 3 has more information about the neighbors, or shows the error message when you have an incorrect command. Note that if there is no neighbor in the batman network, then it will display nothing.
Level 4 has so many information about the batman network, for example, how many packets you sent, and sent to where, or how many packets you got, and received from where etc.

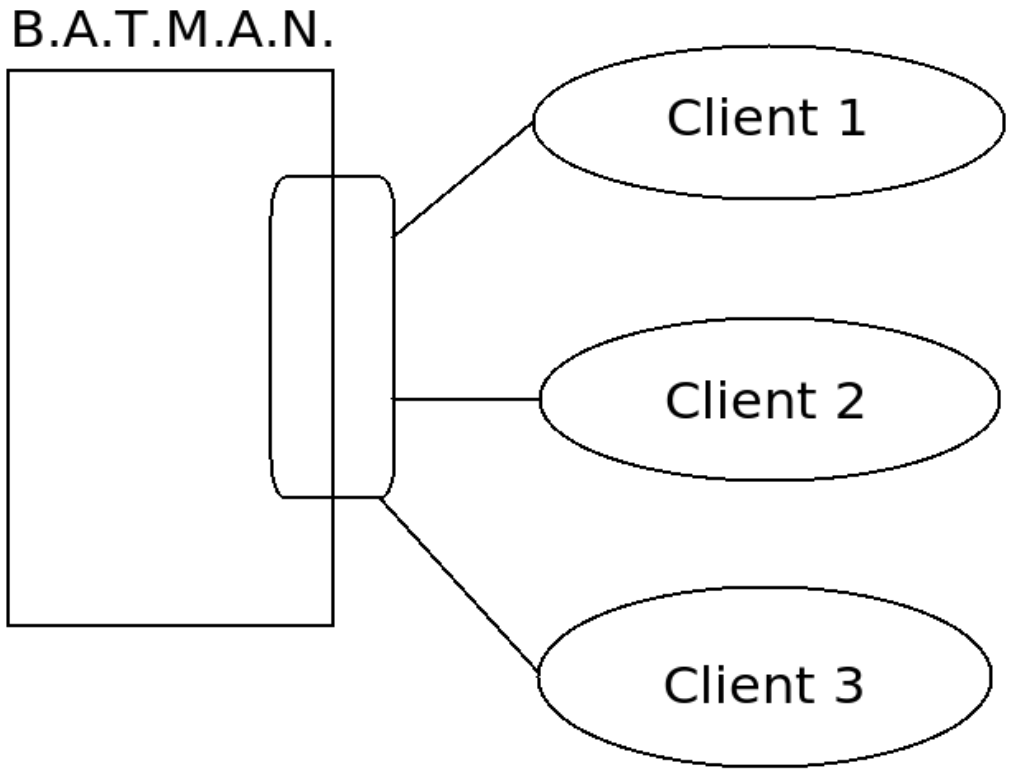
For example, you can run in normal start:

```
$ batmand -d 1 eth1
```

`-c connect via unix socket`

The batman daemon offers a unix socket interface to which you can connect.
First, you have to create a batman daemon on your host, then use -c to connect to its interface.

Note that you can create as many client sockets as you like.




```
$ batmand eth1
$ batmand  -c -d 1
```

In this case, you ask the daemon to output debug level 1 in your current shell. The batman daemon will update the information after a period of time.

Note that if you use -c flag, then you only can use -d to see the debug level.

`-b run connection in batch mode`

The debug information are updated after a period of time by default, so if you use -b it will execute once and then stop.

```
$ batmand eth1
$ batmand  -b -c -d 1
```

In this case, it means run debug level 1 once.

Note that -b can only be used with -c and debug level 1 & 2.

-g gateway class

Gateway class can set eleven values, it means how much bandwidth you want to share.

default: 0 -> this is not an internet gateway
allowed values: 1 -> modem line
2 -> ISDN line
3 -> double ISDN
4 -> 256 KBit
5 -> UMTS / 0.5 MBit
6 -> 1 MBit
7 -> 2 MBit
8 -> 3 MBit
9 -> 5 MBit
10 -> 6 MBit
11 -> >6 Mbit

You only can set the value in a normal start

```
$ batmand -g 7 -d 3  eth1
```

Note that if you use debug level 3, then you will know whether you succeed setting the gateway class or not.

-r routing class

Routing class can set to four values, it means this node in the batman network wants to connect the Internet and chooses its internet gateway based on certain criteria:

default: 0 -> set no default route
allowed values: 1 -> use fast internet connection
2 -> use stable internet connection
3 -> use best statistic internet connection

In level 1, B.A.T.M.A.N tries to find the best available connection by watching the uplinks throughput and the link quality.
In level 2, B.A.T.M.A.N observes the internet nodes and tries to find out which one is the most reliable. This mode is not implemented yet but will follow in batman 0.3.
In level 3, B.A.T.M.A.N only compares the link quality of the internet node and chooses the one with the best connection.

```
$ batmand -r 3 -d 3  eth1
```

In this case, the batman daemon will choose the best statistic internet connection for you. Note that if you use debug level 3, then you will know whether you succeeded setting the routing class or not.

-p preferred gateway

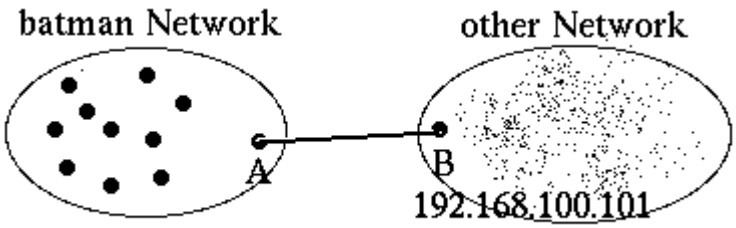
Set the default gateway by yourself.
Note that you have to use -r to tell batman daemon you want to set the default gateway, because -r will be used if the preferred gateway is not available.

```
$ batmand -r 3 -d 3 -p 192.168.1.1 eth1
```

In this case, you set 192.168.1.1 as your preferred gateway, so all of your internet packets will be sent to the 192.168.1.1.

-a announce network(s)

“ announce network “ means a node announces the connection to another network.
For example, if you are the node A, and you can connect to “other network”, then you can execute -a to announce the gateway.



If the other nodes in the batman network want to connect to node B after receiving the announce network information form node A, then they will know they can use node A as gateway to reach node B.

Now, you know what a announced network is, but executing this command is wrong:

```
$ batmand  -a 192.168.100.101  eth1
```

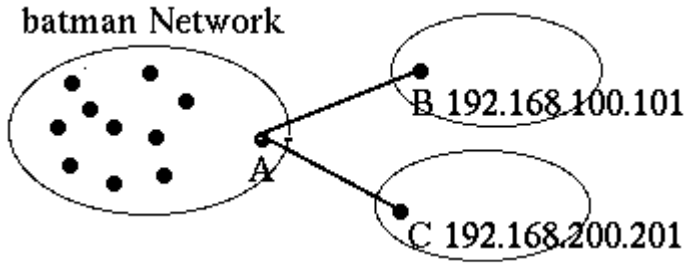
Because you have to specify the netmask parameter and different netmask parameters cause different results. Let's make a example:

```
$ batmand  -a 192.168.100.101/32  eth1
```

In this case, it means that node A can only connect to node B, because your parameter is /32.

```
$ batmand  -a 192.168.100.101/24  eth1
```

In this case, it means that node A can connect to the whole 192.168.100.x network, because your parameter is /24. So, if you use different netmask values, then the results are different.



Node A can announce more than one network. To announce two networks execute the following command:

```
$ batmand  -a 192.168.100.101/24  -a 192.168.200.201/24 eth1
```

Note that node A has to have a route to connect the node or network.

`-s visualization server`

Since no topology database is computed by the protocol an additional solution to create topology graphs has been implemented, the Vis-Server. Batman-daemons may send their local view about their single-hop neighbors to the Vis-server. The Vis-Server collects the information and provides data in a format similar to OLSR's topology information output. Therefore existing solutions to draw topology graphs developed for OLSR can be used to visualize mesh-clouds using B.A.T.M.A.N.

III. Reference link

- http://open-mesh.net/ <== B.A.T.M.A.N
- http://www.olsr.org/ <== OLSR

IV. Troubleshooting

1. Why the batman daemon doesn't reload the setting after I fixed the main IP?

You have to restart the batman daemon after you modified any network configuration, otherwise the batman daemon won't use the new settings.

```
$ killall batmand
$ batmand eth1
```

2. Why I can't connect to the Internet after setting the default gateway?

You have to use NAT on your gateway or firewall if you use the -r or -p options to set default route.

```
$ iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Note that you don't set the default route by yourself.