batman-adv

Batman advanced is a new approach to wireless networking which does no longer operate on the IP basis. Unlike the batman daemon, which exchanges information using UDP packets and sets routing tables, batman-advanced operates on ISO/OSI Layer 2 only and uses and routes (or better: bridges) Ethernet Frames. It emulates a virtual network switch of all nodes participating. Therefore all nodes appear to be link local, thus all higher operating protocols won't be affected by any changes within the network. You can run almost any protocol above batman advanced, prominent examples are: IPv4, IPv6, DHCP, IPX.

Batman advanced was implemented as a Linux kernel driver to reduce the overhead to a minimum. It does not depend on any (other) network driver, and can be used on wifi as well as ethernet lan, vpn, etc ... (anything with ethernet-style layer 2).

Configuration

Load the batman-adv module into your kernel:

```
$ insmod batman-adv.ko
```

The module is now waiting for activation. You must add some interfaces on which batman-adv can operate. The batman-adv soft-interface can be created using the iproute2 tool `ip`:

```
$ ip link add name bat0 type batadv
```

To activate a given interface simply attach it to the `bat0` interface:

```
$ ip link set dev eth0 master bat0
```

Repeat this step for all interfaces you wish to add. Now batman-adv starts using/broadcasting on this/these interface(s).

To deactivate an interface you have to detach it from the "bat0" interface:

```
$ ip link set dev eth0 nomaster
```

The same can also be done using the batctl interface subcommand:

```
batctl -m bat0 interface create
batctl -m bat0 interface add -M eth0
```

To detach eth0 and destroy bat0:

```
batctl -m bat0 interface del -M eth0
batctl -m bat0 interface destroy
```

There are additional settings for each batadv mesh interface, vlan and hardif which can be modified using batctl. Detailed information about this can be found in its manual.

For instance, you can check the current originator interval (value in milliseconds which determines how often batman-adv sends its broadcast packets):

```
$ batctl -M bat0 orig_interval
1000
```

and also change its value:

```
$ batctl -M bat0 orig_interval 3000
```

In very mobile scenarios, you might want to adjust the originator interval to a lower value. This will make the mesh more responsive to topology changes, but will also increase the overhead.

Information about the current state can be accessed via the batadv generic netlink family. batctl provides a human readable version via its debug tables subcommands.

Usage

To make use of your newly created mesh, batman advanced provides a new interface "bat0" which you should use from this point on. All interfaces added to batman advanced are not relevant any longer because batman handles them for you. Basically, one "hands over" the data by using the batman interface and batman will make sure it reaches its destination.

The "bat0" interface can be used like any other regular interface. It needs an IP address which can be either statically configured or dynamically (by using DHCP or similar services):

```
NodeA: ip link set up dev bat0
NodeA: ip addr add 192.168.0.1/24 dev bat0

NodeB: ip link set up dev bat0
NodeB: ip addr add 192.168.0.2/24 dev bat0
```

```
NodeB: ping 192.168.0.1
```

Note: In order to avoid problems remove all IP addresses previously assigned to interfaces now used by batman advanced, e.g.:

```
$ ip addr flush dev eth0
```

Logging/Debugging

All error messages, warnings and information messages are sent to the kernel log. Depending on your operating system distribution this can be read in one of a number of ways. Try using the commands: `dmesg`, `logread`, or looking in the files `/var/log/kern.log` or `/var/log/syslog`. All batman-adv messages are prefixed with "batman-adv:" So to see just these messages try:

```
$ dmesg | grep batman-adv
```

When investigating problems with your mesh network, it is sometimes necessary to see more detailed debug messages. This must be enabled when compiling the batman-adv module. When building batman-adv as part of the kernel, use "make menuconfig" and enable the option `B.A.T.M.A.N. debugging` (`CONFIG_BATMAN_ADV_DEBUG=y`).

Those additional debug messages can be accessed using the perf infrastructure:

```
$ trace-cmd stream -e batadv:batadv_dbg
```

The additional debug output is by default disabled. It can be enabled during run time:

```
$ batctl -m bat0 loglevel routes tt
```

will enable debug messages for when routes and translation table entries change.

Counters for different types of packets entering and leaving the batman-adv module are available through ethtool:

```
$ ethtool --statistics bat0
```

batctl

As batman advanced operates on layer 2, all hosts participating in the virtual switch are completely transparent for all protocols above layer 2. Therefore the common diagnosis tools do not work as expected. To overcome these problems, batctl was created. At the moment the batctl contains ping, traceroute, tcpdump and interfaces to the kernel module settings.

For more information, please see the manpage (`man batctl`).

batctl is available on https://www.open-mesh.org/

Contact

Please send us comments, experiences, questions, anything :)

IRC:
: #batadv on ircs://irc.hackint.org/

Mailing-list:
: b.a.t.m.a.n@open-mesh.org (optional subscription at https://lists.open-mesh.org/mailman3/postorius/lists/b.a.t.m.a.n.lists.open-mesh.org/)

You can also contact the Authors:

- Marek Lindner <mareklindner@neomailbox.ch>
- Simon Wunderlich <sw@simonwunderlich.de>