

NAME

carp - Common Address Redundancy Protocol

SYNOPSIS

device carp

DESCRIPTION

The CARP allows multiple hosts on the same local network to share a set of IPv4 and/or IPv6 addresses. Its primary purpose is to ensure that these addresses are always available.

To use **carp**, the administrator needs to configure at a minimum a common virtual host ID (vhid), and attach at least one IP address to this vhid on each machine which is to take part in the virtual group. Additional parameters can also be set on a per-vhid basis: **advbase** and **advskew**, which are used to control how frequently the host sends advertisements when it is the master for a virtual host, and **pass** which is used to authenticate **carp** advertisements. The **advbase** parameter stands for "advertisement base". It is measured in seconds and specifies the base of the advertisement interval. The **advskew** parameter stands for "advertisement skew". It is measured in 1/256 of seconds. It is added to the base advertisement interval to make one host advertise a bit slower than the other does. Both **advbase** and **advskew** are put inside CARP advertisements. These values can be configured using `ifconfig(8)`, or through the `SIOCSVH ioctl(2)`.

CARP defaults to using multicast messages, but can be configured to unicast announcements to peers using the **peer** and **peer6** parameters. Default addresses can be restored using **mcast** and **mcast6**. Note that TTL verification is disabled if the peer address is not a multicast address. These values can be configured using `ifconfig(8)`, or through the `SIOCSPEER ioctl(2)`.

CARP virtual hosts can be configured on multicast-capable interfaces: Ethernet, layer 2 VLAN, FDDI and Token Ring. An arbitrary number of virtual host IDs can be configured on an interface. An arbitrary number of IPv4 or IPv6 addresses can be attached to a particular vhid. It is important that all hosts participating in a vhid have the same list of prefixes configured on the vhid, since all the prefixes are included in the cryptographic checksum supplied in each advertisement. Multiple vhid's running on one interface participate in master/backup elections independently.

Additionally, there are a number of global parameters which can be set using `sysctl(8)`:

net.inet.carp.allow

Allow **carp** operation. When disabled, virtual hosts remain in initial state, neither sending nor receiving announcements or traffic. Enabled by default.

net.inet.carp.preempt

Allow virtual hosts to preempt each other. When enabled, a vhid

in a backup state would preempt a master that is announcing itself with a lower advskew. Disabled by default.

net.inet.carp.dscp

DSCP value in carp packet. Valid Values are 0 to 63. A value of 4 is equivalent to the old standard of TOS LOW_DELAY. TOS values were deprecated and replaced by DSCP in 1998. The default value is 56 (CS7/Network Control).

net.inet.carp.log

Determines what events relating to **carp** vhids are logged. A value of 0 disables any logging. A value of 1 enables logging state changes of **carp** vhids. Values above 1 enable logging of bad **carp** packets. The default value is 1.

net.inet.carp.demotion

This value shows the current level of CARP demotion. The value is added to the actual advskew sent in announcements for all vhids. During normal system operation the demotion factor is zero. However, problematic conditions raise its level: when **carp** experiences problem with sending announcements, when an interface running a vhid goes down, or while the pfsync(4) interface is not synchronized. The demotion factor can be adjusted writing to the sysctl oid. The signed value supplied to the sysctl(8) command is added to current demotion factor. This allows to control **carp** behaviour depending on some external conditions, for example on the status of some daemon utility.

net.inet.carp.ifdown_demotion_factor

This value is added to *net.inet.carp.demotion* when an interface running a vhid goes down. The default value is 240 (the maximum advskew value).

net.inet.carp.senderr_demotion_factor

This value is added to *net.inet.carp.demotion* when **carp** experiences errors sending its announcements. The default value is 240 (the maximum advskew value).

STATE CHANGE NOTIFICATIONS

Sometimes it is useful to get notified about **carp** status change events. This can be accomplished by using devd(8) hooks. Master/slave events are signalled under system CARP. The subsystem specifies the vhid and name of the interface where the master/slave event occurred. The type of the message displays the new state of the vhid. Please see devd.conf(5) and the *EXAMPLES* section for more information.

EXAMPLES

For firewalls and routers with multiple interfaces, it is desirable to failover all of the addresses running **carp** together, when one of the physical interfaces goes down. This is achieved by the use of the `preempt` option. Enable it on both hosts A and B:

```
sysctl net.inet.carp.preempt=1
```

Assume that host A is the preferred master and we are running the 192.168.1.0/24 prefix on `em0` and 192.168.2.0/24 on `em1`. This is the setup for host A (`advskew` is above 0 so it could be overwritten in the emergency situation from the other host):

```
ifconfig em0 vhid 1 advskew 100 pass mekmitasdigoat 192.168.1.1/24
ifconfig em1 vhid 2 advskew 100 pass mekmitasdigoat 192.168.2.1/24
```

The setup for host B is identical, but it has a higher **advskew**:

```
ifconfig em0 vhid 1 advskew 200 pass mekmitasdigoat 192.168.1.1/24
ifconfig em1 vhid 2 advskew 200 pass mekmitasdigoat 192.168.2.1/24
```

When one of the physical interfaces of host A fails, **advskew** is demoted to a configured value on all its **carp** vhids. Due to the `preempt` option, host B would start announcing itself, and thus preempt host A on both interfaces instead of just the failed one.

Processing of **carp** status change events can be set up by using the following `devd.conf` rule:

```
notify 0 {
    match "system"      "CARP";
    match "subsystem"   "[0-9]+@[0-9a-z.]+";
    match "type"        "(MASTER|BACKUP)";
    action "/root/carpcontrol.sh $subsystem $type";
};
```

To see **carp** packets decoded in `tcpdump(1)` output, one needs to specify the **-T carp** option, otherwise `tcpdump(1)` will interpret them as VRRP packets:

```
tcpdump -npi vlan0 -T carp
```

SEE ALSO

`tcpdump(1)`, `inet(4)`, `pfsync(4)`, `devd.conf(5)`, `rc.conf(5)`, `ifconfig(8)`, `sysctl(8)`

HISTORY

The **carp** device first appeared in OpenBSD 3.5. The **carp** device was imported into FreeBSD 5.4. In FreeBSD 10.0, **carp** was significantly rewritten, and is no longer a pseudo-interface.