MyWebUniversity
python
Rocky Linux
Manual Pages
FPDF Library
PDF generator
FPDF
Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'teamd.conf.5'

$ man teamd.conf.5

TEAMD.CONF(5)   Team daemon configuration   TEAMD.CONF(5)

NAME

teamd.conf ? libteam daemon configuration file

DESCRIPTION

teamd uses JSON format configuration.

OPTIONS

device (string)

Desired name of new team device.

debug_level (int)

Level of debug messages. The higher it is the more debug mes?

sages will be printed. It is the same as adding "-g" command

line options.

Default: 0 (disabled)

hwaddr (string)

Desired hardware address of new team device. Usual MAC address

format is accepted.

runner.name (string)

Name of team device. The following runners are available:

broadcast ? Simple runner which directs the team device to transmit packets via all ports.

roundrobin ? Simple runner which directs the team device to transmits packets in a round-robin fashion.

random ? Simple runner which directs the team device to trans? mits packets on a randomly selected port.

activebackup ? Watches for link changes and selects active port to be used for data transfers.

loadbalance ? To do passive load balancing, runner only sets up BPF hash function which will determine port for packet transmit. To do active load balancing, runner moves hashes among available ports trying to reach perfect balance.

lacp ? Implements 802.3ad LACP protocol. Can use same Tx port selection possibilities as loadbalance runner.

Default: roundrobin

notify_peers.count (int)

Number of bursts of unsolicited NAs and gratuitous ARP packets sent after port is enabled or disabled.

Default: 0 (disabled)

Default for activebackup runner: 1

notify_peers.interval (int)

Value is positive number in milliseconds. Specifies an interval between bursts of notify-peer packets.

Default: 0

mcast_rejoin.count (int)

Number of bursts of multicast group rejoin requests sent after port is enabled or disabled.

Default: 0 (disabled)

Default for activebackup runner: 1

mcast_rejoin.interval (int)

Value is positive number in milliseconds. Specifies an interval between bursts of multicast group rejoin requests.

Default: 0

link_watch.name | ports.PORTIFNAME.link_watch.name (string)

Name of link watcher to be used. The following link watchers are available:

ethtool ? Uses Libteam lib to get port ethtool state changes.

arp_ping ? ARP requests are sent through a port. If an ARP reply is received, the link is considered to be up.

nsna_ping ? Similar to the previous, except that it uses IPv6 Neighbor Solicitation / Neighbor Advertisement mechanism. This is an alternative to arp_ping and becomes handy in pure-IPv6 en? vironments.

ports (object)

List of ports, network devices, to be used in a team device.

See examples for more information.

ports.PORTIFNAME.queue_id (int)

ID of queue which this port should be mapped to.

Default: None

ACTIVE-BACKUP RUNNER SPECIFIC OPTIONS

runner.hwaddr_policy (string)

This defines the policy of how hardware addresses of team device and port devices should be set during the team lifetime. The following are available:

same_all ? All ports will always have the same hardware address as the associated team device.

by_active ? Team device adopts the hardware address of the cur? rently active port. This is useful when the port device is not able to change its hardware address.

only_active ? Only the active port adopts the hardware address of the team device. The others have their own.

Default: same_all

ports.PORTIFNAME.prio (int)

Port priority. The higher number means higher priority.

Default: 0

ports.PORTIFNAME.sticky (bool)

Flag which indicates if the port is sticky. If set, it means the

port does not get unselected if another port with higher prior?

ity or better parameters becomes available.

Default: false

LOAD BALANCE RUNNER SPECIFIC OPTIONS

runner.tx_hash (array)

List of fragment types (strings) which should be used for packet

Tx hash computation. The following are available:

eth ? Uses source and destination MAC addresses.

vlan ? Uses VLAN id.

ipv4 ? Uses source and destination IPv4 addresses.

ipv6 ? Uses source and destination IPv6 addresses.

ip ? Uses source and destination IPv4 and IPv6 addresses.

l3 ? Uses source and destination IPv4 and IPv6 addresses.

tcp ? Uses source and destination TCP ports.

udp ? Uses source and destination UDP ports.

sctp ? Uses source and destination SCTP ports.

l4 ? Uses source and destination TCP and UDP and SCTP ports.

Default: ["eth", "ipv4", "ipv6"]

runner.tx_balancer.name (string)

Name of active Tx balancer. Active Tx balancing is disabled by

default. The only value available is basic.

Default: None

runner.tx_balancer.balancing_interval (int)

In tenths of a second. Periodic interval between rebalancing.

Default: 50

LACP RUNNER SPECIFIC OPTIONS

runner.active (bool)

If active is true LACPDU frames are sent along the configured

links periodically. If not, it acts as "speak when spoken to".

Default: true

runner.fast_rate (bool)

Option specifies the rate at which our link partner is asked to transmit LACPDU packets. If this is true then packets will be

sent once per second. Otherwise they will be sent every 30 sec?

onds.

Default: false

runner.tx_hash (array)

Same as for load balance runner.

runner.tx_balancer.name (string)

Same as for load balance runner.

runner.tx_balancer.balancing_interval (int)

Same as for load balance runner.

runner.sys_prio (int)

System priority, value can be 0 ? 65535.

Default: 65535

runner.min_ports (int)

Specifies the minimum number of ports that must be active before

asserting carrier in the master interface, value can be 1 ? 255.

Default: 1

runner.agg_select_policy (string)

This selects the policy of how the aggregators will be selected.

The following are available:

lacp_prio ? Aggregator with highest priority according to LACP

standard will be selected. Aggregator priority is affected by

per-port option lacp_prio.

lacp_prio_stable ? Same as previous one, except do not replace

selected aggregator if it is still usable.

bandwidth ? Select aggregator with highest total bandwidth.

count ? Select aggregator with highest number of ports.

port_config ? Aggregator with highest priority according to per-

port options prio and sticky will be selected. This means that

the aggregator containing the port with the highest priority

will be selected unless at least one of the ports in the cur?

rently selected aggregator is sticky.

Default: lacp_prio

ports.PORTIFNAME.lacp_prio (int)

Port priority according to LACP standard. The lower number means higher priority.

Default: 255

ports.PORTIFNAME.lacp_key (int)

Port key according to LACP standard. It is only possible to ag?gregate ports with the same key.

Default: 0

ETHTOOL LINK WATCH SPECIFIC OPTIONS

link_watch.delay_up | ports.PORTIFNAME.link_watch.delay_up (int)

Value is a positive number in milliseconds. It is the delay be?tween the link coming up and the runner being notified about it.

Default: 0

link_watch.delay_down | ports.PORTIFNAME.link_watch.delay_down (int)

Value is a positive number in milliseconds. It is the delay be?tween the link going down and the runner being notified about it.

Default: 0

ARP PING LINK WATCH SPECIFIC OPTIONS

link_watch.interval | ports.PORTIFNAME.link_watch.interval (int)

Value is a positive number in milliseconds. It is the interval between ARP requests being sent.

Default: 1000

link_watch.init_wait | ports.PORTIFNAME.link_watch.init_wait (int)

Value is a positive number in milliseconds. It is the delay be?tween link watch initialization and the first ARP request being sent.

Default: 0

link_watch.missed_max | ports.PORTIFNAME.link_watch.missed_max (int)

Maximum number of missed ARP replies. If this number is ex?ceeded, link is reported as down.

Default: 3

link_watch.source_host | ports.PORTIFNAME.link_watch.source_host (host?

name)

Hostname to be converted to IP address which will be filled into

ARP request as source address.

Default: 0.0.0.0

link_watch.target_host | ports.PORTIFNAME.link_watch.target_host (host?

name)

Hostname to be converted to IP address which will be filled into

ARP request as destination address.

link_watch.validate_active | ports.PORTIFNAME.link_watch.validate_ac?

tive (bool)

Validate received ARP packets on active ports. If this is not

set, all incoming ARP packets will be considered as a good re?

ply.

Default: false

link_watch.validate_inactive | ports.PORTIFNAME.link_watch.validate_in?

active (bool)

Validate received ARP packets on inactive ports. If this is not

set, all incoming ARP packets will be considered as a good re?

ply.

Default: false

link_watch.vlanid | ports.PORTIFNAME.link_watch.vlanid (int)

By default, ARP requests are sent without VLAN tags. This option

causes outgoing ARP requests to be sent with the specified VLAN

ID number.

Default: None

link_watch.send_always | ports.PORTIFNAME.link_watch.send_always (bool)

By default, ARP requests are sent on active ports only. This op?

tion allows sending even on inactive ports.

Default: false

NS/NA PING LINK WATCH SPECIFIC OPTIONS

link_watch.interval | ports.PORTIFNAME.link_watch.interval (int)

Value is a positive number in milliseconds. It is the interval

between sending NS packets.

Default: 1000

link_watch.init_wait | ports.PORTIFNAME.link_watch.init_wait (int)

Value is a positive number in milliseconds. It is the delay be?
tween link watch initialization and the first NS packet being
sent.

link_watch.missed_max | ports.PORTIFNAME.link_watch.missed_max (int)

Maximum number of missed NA reply packets. If this number is ex?
ceeded, link is reported as down.

Default: 3

link_watch.target_host | ports.PORTIFNAME.link_watch.target_host (host?
name)

Hostname to be converted to IPv6 address which will be filled
into NS packet as target address.

EXAMPLES

```
{
  "device": "team0",
  "runner": {"name": "roundrobin"},
  "ports": {"eth1": {}, "eth2": {}}
}
```

Very basic configuration.

```
{
  "device": "team0",
  "runner": {"name": "activebackup"},
  "link_watch": {"name": "ethtool"},
  "ports": {
    "eth1": {
      "prio": -10,
      "sticky": true
    },
    "eth2": {
      "prio": 100
    }
  }
}
```

This configuration uses active-backup runner with ethtool link watcher.

Port eth2 has higher priority, but the sticky flag ensures that if eth1

becomes active, it stays active while the link remains up.

```
{
  "device": "team0",
  "runner": {"name": "activebackup"},
  "link_watch": {
    "name": "ethtool",
    "delay_up": 2500,
    "delay_down": 1000
  },
  "ports": {
    "eth1": {
      "prio": -10,
      "sticky": true
    },
    "eth2": {
      "prio": 100
    }
  }
}
```

Similar to the previous one. Only difference is that link changes are

not propagated to the runner immediately, but delays are applied.

```
{
  "device": "team0",
  "runner": {"name": "activebackup"},
  "link_watch":     {
    "name": "arp_ping",
    "interval": 100,
    "missed_max": 30,
    "target_host": "192.168.23.1"
  },
  "ports": {
    "eth1": {
      "prio": -10,
      "sticky": true
    },
    "eth2": {
      "prio": 100
    }
  }
}
```

This configuration uses ARP ping link watch.

```
{
"device": "team0",
"runner": {"name": "activebackup"},
"link_watch": [
  {
    "name": "arp_ping",
    "interval": 100,
    "missed_max": 30,
    "target_host": "192.168.23.1"
  },
  {
    "name": "arp_ping",
    "interval": 50,
    "missed_max": 20,
    "target_host": "192.168.24.1"
  }
],
"ports": {
  "eth1": {
    "prio": -10,
    "sticky": true
  },
  "eth2": {
    "prio": 100
    }
  }
}
```

Similar to the previous one, only this time two link watchers are used

at the same time.

```
{
  "device": "team0",
  "runner": {
    "name": "loadbalance",
    "tx_hash": ["eth", "ipv4", "ipv6"]
  },
  "ports": {"eth1": {}, "eth2": {}}
}
```

Configuration for hash-based passive Tx load balancing.

```
{
  "device": "team0",
  "runner": {
    "name": "loadbalance",
    "tx_hash": ["eth", "ipv4", "ipv6"],
    "tx_balancer": {
      "name": "basic"
    }
  },
  "ports": {"eth1": {}, "eth2": {}}
}
```

Configuration for active Tx load balancing using basic load balancer.

```
{
  "device": "team0",
  "runner": {
    "name": "lacp",
    "active": true,
    "fast_rate": true,
    "tx_hash": ["eth", "ipv4", "ipv6"]
  },
  "link_watch": {"name": "ethtool"},
  "ports": {"eth1": {}, "eth2": {}}
}
```

Configuration for connection to LACP capable counterpart.

SEE ALSO

teamd(8), teamdctl(8), teamnl(8), bond2team(1)

AUTHOR

Jiri Pirko is the original author and current maintainer of libteam.