Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'bridge.8' command

$ man bridge.8

BRIDGE(8) Linux BRIDGE(8)

NAME

bridge - show / manipulate bridge addresses and devices

SYNOPSIS

bridge [OPTIONS] OBJECT { COMMAND | help }

OBJECT := { link | fdb | mdb | vlan | vni | monitor }

OPTIONS := { -V[ersion] | -s[tatistics] | -n[etns] name | -b[atch]
filename | -c[olor] | -p[retty] | -j[son] | -o[neline] }

bridge link set dev DEV [cost COST] [priority PRIO] [state STATE]
[guard { on | off }] [hairpin { on | off }] [fastleave {
on | off }] [root_block { on | off }] [learning { on | off
}] [learning_sync { on | off }] [flood { on | off }] [hw?
mode { vepa | veb }] [bcast_flood { on | off }] [
mcast_flood { on | off }] [mcast_router MULTICAST_ROUTER] [
mcast_to_unicast { on | off }] [neigh_suppress { on | off }]
[vlan_tunnel { on | off }] [isolated { on | off }] [locked
{ on | off }] [backup_port DEVICE] [nobackup_port] [self
] [master]

bridge link [show] [dev DEV]

bridge fdb { add | append | del | replace } LLADDR dev DEV { local |
static | dynamic } [self] [master] [router] [use] [ex?
tern_learn] [sticky] [src_vni VNI] { [dst IPADDR] [vni
VNI] [port PORT] [via DEVICE] | nhid NHID }

bridge fdb [[show] [br BRDEV] [brport DEV] [vlan VID] [state STATE] [dynamic]]

bridge fdb get [to] LLADDR [br BRDEV] { brport | dev } DEV [vlan VID] [vni VNI] [self] [master] [dynamic]

bridge fdb flush dev DEV [brport DEV] [vlan VID] [self] [master] [[no]permanent | [no]static | [no]dynamic] [[no]added_by_user] [[no]extern_learn] [[no]sticky] [[no]offloaded]

bridge mdb { add | del } dev DEV port PORT grp GROUP [src SOURCE] [permanent | temp] [vid VID]

bridge mdb show [dev DEV]

bridge vlan { add | del } dev DEV vid VID [tunnel_info TUNNEL_ID] [pvid] [untagged] [self] [master]

bridge vlan set dev DEV vid VID [state STP_STATE] [mcast_router MUL? TICAST_ROUTER]

bridge vlan [show | tunnelshow] [dev DEV]

bridge vlan global set dev DEV vid VID [mcast_snooping MULTI? CAST_SNOOPING] [mcast_querier MULTICAST_QUERIER] [mcast_igmp_version IGMP_VERSION] [mcast_mld_version MLD_VER? SION] [mcast_last_member_count LAST_MEMBER_COUNT] [mcast_last_member_interval LAST_MEMBER_INTERVAL] [mcast_startup_query_count STARTUP_QUERY_COUNT] [mcast_startup_query_interval STARTUP_QUERY_INTERVAL] [mcast_membership_interval MEMBERSHIP_INTERVAL] [mcast_querier_interval QUERIER_INTERVAL] [mcast_query_inter? val QUERY_INTERVAL] [mcast_query_response_interval QUERY_RE? SPONSE_INTERVAL]

bridge vlan global [show] [dev DEV] [vid VID]

bridge vlan show [dev DEV]

bridge vni { add | del } dev DEV vni VNI [{ group | remote } IPADDR]

bridge vni show [dev DEV]

bridge monitor [all | neigh | link | mdb | vlan]

OPTIONS

-V, -Version

print the version of the bridge utility and exit.

-s, -stats, -statistics

output more information. If this option is given multiple times,

the amount of information increases. As a rule, the information

is statistics or some time values.

-d, -details

print detailed information about bridge vlan filter entries or

MDB router ports.

-n, -net, -netns <NETNS>

switches bridge to the specified network namespace NETNS. Actu?

ally it just simplifies executing of:

ip netns exec NETNS bridge [OPTIONS] OBJECT { COMMAND | help }

to

bridge -n[etns] NETNS [OPTIONS] OBJECT { COMMAND | help }

-b, -batch <FILENAME>

Read commands from provided file or standard input and invoke

them. First failure will cause termination of bridge command.

-force Don't terminate bridge command on errors in batch mode. If

there were any errors during execution of the commands, the ap?

plication return code will be non zero.

-c[color][={always|auto|never}

Configure color output. If parameter is omitted or always, color

output is enabled regardless of stdout state. If parameter is

auto, stdout is checked to be a terminal before enabling color

output. If parameter is never, color output is disabled. If

specified multiple times, the last one takes precedence. This

flag is ignored if -json is also given.

-j, -json

Output results in JavaScript Object Notation (JSON).

-p, -pretty

When combined with -j generate a pretty JSON output.

-o, -oneline

output each record on a single line, replacing line feeds with the '\' character. This is convenient when you want to count records with wc(1) or to grep(1) the output.

BRIDGE - COMMAND SYNTAX

OBJECT

link - Bridge port.

fdb - Forwarding Database entry.

mdb - Multicast group database entry.

vlan - VLAN filter list.

vni - VNI filter list.

COMMAND

Specifies the action to perform on the object. The set of possible ac? tions depends on the object type. As a rule, it is possible to add, delete and show (or list) objects, but some objects do not allow all of these operations or have some additional commands. The help command is available for all objects. It prints out a list of available com? mands and argument syntax conventions.

If no command is given, some default command is assumed. Usually it is list or, if the objects of this class cannot be listed, help.

bridge link - bridge port

link objects correspond to the port devices of the bridge.

The corresponding commands set and display port status and bridge spe? cific attributes.

bridge link set - set bridge specific attributes on a port

dev NAME

interface name of the bridge port

cost COST

the STP path cost of the specified port.

priority PRIO

the STP port priority. The priority value is an unsigned 8-bit quantity (number between 0 and 255). This metric is used in the designated port an droot port selection algorithms.

state STATE

the operation state of the port. Except state 0 (disable STP or BPDU filter feature), this is primarily used by user space STP/RSTP implementation. One may enter port state name (case in? sensitive), or one of the numbers below. Negative inputs are ig? nored, and unrecognized names return an error.

0 - port is in STP DISABLED state. Make this port completely in? active for STP. This is also called BPDU filter and could be used to disable STP on an untrusted port, like a leaf virtual devices.

1 - port is in STP LISTENING state. Only valid if STP is enabled on the bridge. In this state the port listens for STP BPDUs and drops all other traffic frames.

2 - port is in STP LEARNING state. Only valid if STP is enabled on the bridge. In this state the port will accept traffic only for the purpose of updating MAC address tables.

3 - port is in STP FORWARDING state. Port is fully active.

4 - port is in STP BLOCKING state. Only valid if STP is enabled on the bridge. This state is used during the STP election process. In this state, port will only process STP BPDUs.

guard on or guard off

Controls whether STP BPDUs will be processed by the bridge port. By default, the flag is turned off allowed BPDU processing. Turning this flag on will disables the bridge port if a STP BPDU packet is received.

If running Spanning Tree on bridge, hostile devices on the net? work may send BPDU on a port and cause network failure. Setting guard on will detect and stop this by disabling the port. The port will be restarted if link is brought down, or removed and reattached. For example if guard is enable on eth0:

ip link set dev eth0 down; ip link set dev eth0 up

hairpin on or hairpin off

Controls whether traffic may be send back out of the port on which it was received. This option is also called reflective re?

lay mode, and is used to support basic VEPA (Virtual Ethernet Port Aggregator) capabilities. By default, this flag is turned off and the bridge will not forward traffic back out of the re? ceiving port.

fastleave on or fastleave off

This flag allows the bridge to immediately stop multicast traf? fic on a port that receives IGMP Leave message. It is only used with IGMP snooping is enabled on the bridge. By default the flag is off.

root_block on or root_block off

Controls whether a given port is allowed to become root port or not. Only used when STP is enabled on the bridge. By default the flag is off.

This feature is also called root port guard. If BPDU is re? ceived from a leaf (edge) port, it should not be elected as root port. This could be used if using STP on a bridge and the down? stream bridges are not fully trusted; this prevents a hostile guest from rerouting traffic.

learning on or learning off

Controls whether a given port will learn MAC addresses from re? ceived traffic or not. If learning if off, the bridge will end up flooding any traffic for which it has no FDB entry. By de? fault this flag is on.

learning_sync on or learning_sync off

Controls whether a given port will sync MAC addresses learned on device port to bridge FDB.

flood on or flood off

Controls whether unicast traffic for which there is no FDB entry will be flooded towards this given port. By default this flag is on.

hwmode Some network interface cards support HW bridge functionality and they may be configured in different modes. Currently support modes are:

vepa - Data sent between HW ports is sent on the wire to the ex?

ternal switch.

veb - bridging happens in hardware.

bcast_flood on or bcast_flood off

Controls flooding of broadcast traffic on the given port. By

default this flag is on.

mcast_flood on or mcast_flood off

Controls whether multicast traffic for which there is no MDB en?

try will be flooded towards this given port. By default this

flag is on.

mcast_router MULTICAST_ROUTER

This flag is almost the same as the per-VLAN flag, see below,

except its value can only be set in the range 0-2. The default

is 1 where the bridge figures out automatically where an

IGMP/MLD querier, MRDISC capable device, or PIM router, is lo?

cated. Setting this flag to 2 is useful in cases where the mul?

ticast router does not indicate its presence in any meaningful

way (e.g. older versions of SMCRoute, or mrouted), or when there

is a need for forwarding both known and unknown IP multicast to

a secondary/backup router.

mcast_to_unicast on or mcast_to_unicast off

Controls whether a given port will replicate packets using uni?

cast instead of multicast. By default this flag is off.

This is done by copying the packet per host and changing the

multicast destination MAC to a unicast one accordingly.

mcast_to_unicast works on top of the multicast snooping feature

of the bridge. Which means unicast copies are only delivered to

hosts which are interested in it and signalized this via

IGMP/MLD reports previously.

This feature is intended for interface types which have a more

reliable and/or efficient way to deliver unicast packets than

broadcast ones (e.g. WiFi).

However, it should only be enabled on interfaces where no

IGMPv2/MLDv1 report suppression takes place. IGMP/MLD report suppression issue is usually overcome by the network daemon (supplicant) enabling AP isolation and by that separating all STAs.

Delivery of STA-to-STA IP multicast is made possible again by enabling and utilizing the bridge hairpin mode, which considers the incoming port as a potential outgoing port, too (see hairpin option). Hairpin mode is performed after multicast snooping, therefore leading to only deliver reports to STAs running a mul? ticast router.

neigh_suppress on or neigh_suppress off

Controls whether neigh discovery (arp and nd) proxy and suppres? sion is enabled on the port. By default this flag is off.

vlan_tunnel on or vlan_tunnel off

Controls whether vlan to tunnel mapping is enabled on the port. By default this flag is off.

isolated on or isolated off

Controls whether a given port will be isolated, which means it will be able to communicate with non-isolated ports only. By default this flag is off.

locked on or locked off

Controls whether a port will be locked, meaning that hosts be? hind the port will not be able to communicate through the port unless an FDB entry with the units MAC address is in the FDB. The common use is that hosts are allowed access through authen? tication with the IEEE 802.1X protocol or based on whitelists or like setups. By default this flag is off.

backup_port DEVICE

If the port loses carrier all traffic will be redirected to the configured backup port

nobackup_port

Removes the currently configured backup port

self link setting is configured on specified physical device

master link setting is configured on the software bridge (default)

-t, -timestamp

display current time when using monitor option.

bridge link show - list ports configuration for all bridges.

This command displays port configuration and flags for all bridges.

To display port configuration and flags for a specific bridge, use the

"ip link show master <bridge_device>" command.

bridge fdb - forwarding database management

fdb objects contain known Ethernet addresses on a link.

The corresponding commands display fdb entries, add new entries, append

entries, and delete old ones.

bridge fdb add - add a new fdb entry

This command creates a new fdb entry.

LLADDR the Ethernet MAC address.

dev DEV

the interface to which this address is associated.

local - is a local permanent fdb entry, which means that the

bridge will not forward frames with this destination MAC address

and VLAN ID, but terminate them locally. This flag is default

unless "static" or "dynamic" are explicitly specified.

permanent - this is a synonym for "local"

static - is a static (no arp) fdb entry

dynamic - is a dynamic reachable age-able fdb entry

self - the operation is fulfilled directly by the driver for the

specified network device. If the network device belongs to a

master like a bridge, then the bridge is bypassed and not noti?

fied of this operation (and if the device does notify the

bridge, it is driver-specific behavior and not mandated by this

flag, check the driver for more details). The "bridge fdb add"

command can also be used on the bridge device itself, and in

this case, the added fdb entries will be locally terminated (not

forwarded). In the latter case, the "self" flag is mandatory.

The flag is set by default if "master" is not specified.

master - if the specified network device is a port that belongs to a master device such as a bridge, the operation is fulfilled by the master device's driver, which may in turn notify the port driver too of the address. If the specified device is a master itself, such as a bridge, this flag is invalid.

router - the destination address is associated with a router. Valid if the referenced device is a VXLAN type device and has route short circuit enabled.

use - the address is in use. User space can use this option to indicate to the kernel that the fdb entry is in use.

extern_learn - this entry was learned externally. This option can be used to indicate to the kernel that an entry was hardware or user-space controller learnt dynamic entry. Kernel will not age such an entry.

sticky - this entry will not change its port due to learning.

The next command line parameters apply only when the specified device DEV is of type VXLAN.

dst IPADDR

the IP address of the destination VXLAN tunnel endpoint where the Ethernet MAC ADDRESS resides.

src_vni VNI

the src VNI Network Identifier (or VXLAN Segment ID) this entry belongs to. Used only when the vxlan device is in external or collect metadata mode. If omitted the value specified at vxlan device creation will be used.

vni VNI

the VXLAN VNI Network Identifier (or VXLAN Segment ID) to use to connect to the remote VXLAN tunnel endpoint. If omitted the value specified at vxlan device creation will be used.

port PORT

the UDP destination PORT number to use to connect to the remote VXLAN tunnel endpoint. If omitted the default value is used.

via DEVICE

device name of the outgoing interface for the VXLAN device

driver to reach the remote VXLAN tunnel endpoint.

nhid NHID

ecmp nexthop group for the VXLAN device driver to reach remote

VXLAN tunnel endpoints.

bridge fdb append - append a forwarding database entry

This command adds a new fdb entry with an already known LLADDR. Valid

only for multicast link layer addresses. The command adds support for

broadcast and multicast Ethernet MAC addresses. The Ethernet MAC ad?

dress is added multiple times into the forwarding database and the

vxlan device driver sends a copy of the data packet to each entry

found.

The arguments are the same as with bridge fdb add.

bridge fdb delete - delete a forwarding database entry

This command removes an existing fdb entry.

The arguments are the same as with bridge fdb add.

bridge fdb replace - replace a forwarding database entry

If no matching entry is found, a new one will be created instead.

The arguments are the same as with bridge fdb add.

bridge fdb show - list forwarding entries.

This command displays the current forwarding table.

With the -statistics option, the command becomes verbose. It prints out

the last updated and last used time for each entry.

bridge fdb get - get bridge forwarding entry.

lookup a bridge forwarding table entry.

LLADDR the Ethernet MAC address.

dev DEV

the interface to which this address is associated.

brport DEV

the bridge port to which this address is associated. same as dev

above.

br DEV the bridge to which this address is associated.

self - the address is associated with the port drivers fdb. Usually

hardware.

master - the address is associated with master devices fdb. Usually

software (default).

bridge fdb flush - flush bridge forwarding table entries.

flush the matching bridge forwarding table entries. Some options below have a negated form when "no" is prepended to them (e.g. permanent and nopermanent).

dev DEV

the target device for the operation. If the device is a bridge port and "master" is set then the operation will be fulfilled by its master device's driver and all entries pointing to that port will be deleted.

brport DEV

the target bridge port for the operation. If the bridge device is specified then only entries pointing to the bridge itself will be deleted. Note that the target device specified by this option will override the one specified by dev above.

vlan VID

the target VLAN ID for the operation. Match forwarding table en? tries only with the specified VLAN ID.

self the operation is fulfilled directly by the driver for the speci? fied network device. If the network device belongs to a master like a bridge, then the bridge is bypassed and not notified of this operation. The "bridge fdb flush" command can also be used on the bridge device itself. The flag is set by default if "mas? ter" is not specified.

master if the specified network device is a port that belongs to a mas? ter device such as a bridge, the operation is fulfilled by the master device's driver.

[no]permanent

if specified then only permanent entries will be deleted or re? spectively if "no" is prepended then only non-permanent entries will be deleted.

[no]static

if specified then only static entries will be deleted or respec?

tively if "no" is prepended then only non-static entries will be

deleted.

[no]dynamic

if specified then only dynamic entries will be deleted or re?

spectively if "no" is prepended then only non-dynamic (static or

permanent) entries will be deleted.

[no]added_by_user

if specified then only entries with added_by_user flag will be

deleted or respectively if "no" is prepended then only entries

without added_by_user flag will be deleted.

[no]extern_learn

if specified then only entries with extern_learn flag will be

deleted or respectively if "no" is prepended then only entries

without extern_learn flag will be deleted.

[no]sticky

if specified then only entries with sticky flag will be deleted

or respectively if "no" is prepended then only entries without

sticky flag will be deleted.

[no]offloaded

if specified then only entries with offloaded flag will be

deleted or respectively if "no" is prepended then only entries

without offloaded flag will be deleted.

bridge mdb - multicast group database management

mdb objects contain known IP or L2 multicast group addresses on a link.

The corresponding commands display mdb entries, add new entries, and

delete old ones.

bridge mdb add - add a new multicast group database entry

This command creates a new mdb entry.

dev DEV

the interface where this group address is associated.

port PORT

the port whose link is known to have members of this multicast group.

grp GROUP

the multicast group address (IPv4, IPv6 or L2 multicast) whose members reside on the link connected to the port.

permanent - the mdb entry is permanent. Optional for IPv4 and IPv6, mandatory for L2.

temp - the mdb entry is temporary (default)

src SOURCE

optional source IP address of a sender for this multicast group. If IGMPv3 for IPv4, or MLDv2 for IPv6 respectively, are enabled it will be included in the lookup when forwarding multicast traffic.

vid VID

the VLAN ID which is known to have members of this multicast group.

bridge mdb delete - delete a multicast group database entry

This command removes an existing mdb entry.

The arguments are the same as with bridge mdb add.

bridge mdb show - list multicast group database entries

This command displays the current multicast group membership table. The table is populated by IGMP and MLD snooping in the bridge driver auto? matically. It can be altered by bridge mdb add and bridge mdb del com? mands manually too.

dev DEV

the interface only whose entries should be listed. Default is to list all bridge interfaces.

With the -details option, the command becomes verbose. It prints out the ports known to have a connected router.

With the -statistics option, the command displays timer values for mdb and router port entries.

bridge vlan - VLAN filter list

vlan objects contain known VLAN IDs for a link.

The corresponding commands display vlan filter entries, add new en?

tries, and delete old ones.

bridge vlan add - add a new vlan filter entry

This command creates a new vlan filter entry.

dev NAME

the interface with which this vlan is associated.

vid VID

the VLAN ID that identifies the vlan.

tunnel_info TUNNEL_ID

the TUNNEL ID that maps to this vlan. The tunnel id is set in

dst_metadata for every packet that belongs to this vlan (appli?

cable to bridge ports with vlan_tunnel flag set).

pvid the vlan specified is to be considered a PVID at ingress. Any

untagged frames will be assigned to this VLAN.

untagged

the vlan specified is to be treated as untagged on egress.

self the vlan is configured on the specified physical device. Re?

quired if the device is the bridge device.

master the vlan is configured on the software bridge (default).

bridge vlan delete - delete a vlan filter entry

This command removes an existing vlan filter entry.

The arguments are the same as with bridge vlan add. The pvid and un?

tagged flags are ignored.

bridge vlan set - change vlan filter entry's options

This command changes vlan filter entry's options.

dev NAME

the interface with which this vlan is associated.

vid VID

the VLAN ID that identifies the vlan.

state STP_STATE

the operation state of the vlan. One may enter STP state name

(case insensitive), or one of the numbers below. Negative inputs

are ignored, and unrecognized names return an error. Note that

the state is set only for the vlan of the specified device, e.g.
if it is a bridge port then the state will be set only for the
vlan of the port.
0 - vlan is in STP DISABLED state. Make this vlan completely in?
active for STP. This is also called BPDU filter and could be
used to disable STP on an untrusted vlan.
1 - vlan is in STP LISTENING state. Only valid if STP is enabled
on the bridge. In this state the vlan listens for STP BPDUs and
drops all other traffic frames.
2 - vlan is in STP LEARNING state. Only valid if STP is enabled
on the bridge. In this state the vlan will accept traffic only
for the purpose of updating MAC address tables.
3 - vlan is in STP FORWARDING state. This is the default vlan
state.
4 - vlan is in STP BLOCKING state. Only valid if STP is enabled
on the bridge. This state is used during the STP election
process. In this state, the vlan will only process STP BPDUs.

mcast_router MULTICAST_ROUTER
configure this vlan and interface's multicast router mode, note
that only modes 0 - 2 are available for bridge devices. A vlan
and interface with a multicast router will receive all multicast
traffic. MULTICAST_ROUTER may be either
0 - to disable multicast router.
1 - to let the system detect the presence of routers (default).
2 - to permanently enable multicast traffic forwarding on this
vlan and interface.
3 - to temporarily mark this vlan and port as having a multicast
router, i.e. enable multicast traffic forwarding. This mode is
available only for ports.

bridge vlan show - list vlan configuration.
This command displays the current VLAN filter table.
With the -details option, the command becomes verbose. It displays the
per-vlan options.

With the -statistics option, the command displays per-vlan traffic sta?

tistics.

bridge vlan tunnelshow - list vlan tunnel mapping.

This command displays the current vlan tunnel info mapping.

bridge vlan global set - change vlan filter entry's global options

This command changes vlan filter entry's global options.

dev NAME

the interface with which this vlan is associated. Only bridge

devices are supported for global options.

vid VID

the VLAN ID that identifies the vlan.

mcast_snooping MULTICAST_SNOOPING

turn multicast snooping for VLAN entry with VLAN ID on (MULTI?

CAST_SNOOPING > 0) or off (MULTICAST_SNOOPING == 0). Default is

on.

mcast_querier MULTICAST_QUERIER

enable (MULTICAST_QUERIER > 0) or disable (MULTICAST_QUERIER ==

0) IGMP/MLD querier, ie sending of multicast queries by the

bridge. Default is disabled.

mcast_igmp_version IGMP_VERSION

set the IGMP version. Default is 2.

mcast_mld_version MLD_VERSION

set the MLD version. Default is 1.

mcast_last_member_count LAST_MEMBER_COUNT

set multicast last member count, ie the number of queries the

bridge will send before stopping forwarding a multicast group

after a "leave" message has been received. Default is 2.

mcast_last_member_interval LAST_MEMBER_INTERVAL

interval between queries to find remaining members of a group,

after a "leave" message is received.

mcast_startup_query_count STARTUP_QUERY_COUNT

set the number of queries to send during startup phase. Default

is 2.

mcast_startup_query_interval STARTUP_QUERY_INTERVAL

interval between queries in the startup phase.

mcast_membership_interval MEMBERSHIP_INTERVAL

delay after which the bridge will leave a group, if no member?

ship reports for this group are received.

mcast_querier_interval QUERIER_INTERVAL

interval between queries sent by other routers. If no queries

are seen after this delay has passed, the bridge will start to

send its own queries (as if mcast_querier was enabled).

mcast_query_interval QUERY_INTERVAL

interval between queries sent by the bridge after the end of the

startup phase.

mcast_query_response_interval QUERY_RESPONSE_INTERVAL

set the Max Response Time/Maximum Response Delay for IGMP/MLD

queries sent by the bridge.

bridge vlan global show - list global vlan options.

This command displays the global VLAN options for each VLAN entry.

dev DEV

the interface only whose VLAN global options should be listed.

Default is to list all bridge interfaces.

vid VID

the VLAN ID only whose global options should be listed. Default

is to list all vlans.

bridge vni - VNI filter list

vni objects contain known VNI IDs for a dst metadata vxlan link.

The corresponding commands display vni filter entries, add new entries,

and delete old ones.

bridge vni add - add a new vni filter entry

This command creates a new vni filter entry.

dev NAME

the interface with which this vni is associated.

vni VNI

the VNI ID that identifies the vni.

remote IPADDR

specifies the unicast destination IP address to use in outgoing packets when the destination link layer address is not known in the VXLAN device forwarding database. This parameter cannot be specified with the group.

group IPADDR

specifies the multicast IP address to join for this VNI

bridge vni del - delete a new vni filter entry

This command removes an existing vni filter entry.

The arguments are the same as with bridge vni add.

bridge vni show - list vni filtering configuration.

This command displays the current vni filter table.

With the -statistics option, the command displays per-vni traffic sta? tistics.

dev NAME

shows vni filtering table associated with the vxlan device

bridge monitor - state monitoring

The bridge utility can monitor the state of devices and addresses con? tinuously. This option has a slightly different format. Namely, the monitor command is the first in the command line and then the object list follows:

bridge monitor [all | OBJECT-LIST]

OBJECT-LIST is the list of object types that we want to monitor. It may contain link, fdb, vlan and mdb. If no file argument is given, bridge opens RTNETLINK, listens on it and dumps state changes in the format described in previous sections.

If a file name is given, it does not listen on RTNETLINK, but opens the file containing RTNETLINK messages saved in binary format and dumps them.

NOTES

This command uses facilities added in Linux 3.0.

Although the forwarding table is maintained on a per-bridge device ba? sis the bridge device is not part of the syntax. This is a limitation

of the underlying netlink neighbour message protocol. When displaying the forwarding table, entries for all bridges are displayed.

Add/delete/modify commands determine the underlying bridge device based on the bridge to which the corresponding ethernet device is attached.

SEE ALSO

ip(8)

BUGS

Please direct bugreports and patches to: <netdev@vger.kernel.org>

AUTHOR

Original Manpage by Stephen Hemminger