MCAST-PA

1.0.0

Generated by Doxygen 1.8.6

Tue Jul 17 2018 09:46:41

Contents

Chapter 1

Multicast Packet Accelerator

1.1 Overview

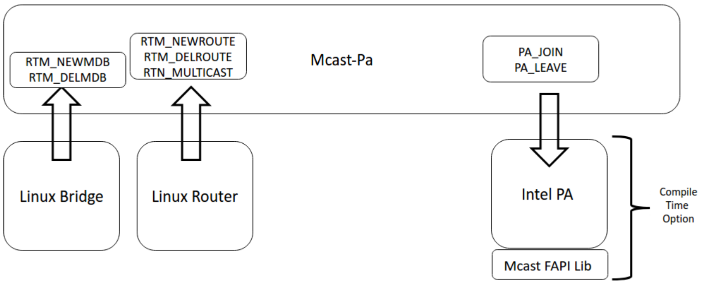
The basic OWRT multicast model consisting of the multicast aware Linux bridge and MCPROXY works on platforms where the CPU has the horsepower to meet the multicast delivery requirements and perform general CPE management functions in parallel.

On platforms where a proprietary acceleration module is required the MCPROXY/bridge model must be augmented with an agent to support insertion and deletion of multicast entries into the acceleration module. This document describes the MCAST-PA agent that augments the MCPROXY/bridge model.

MCAST-PA manages the insertion and deletion of multicast groups into a proprietary multicast accelerator module. Fundamentally MCAST-PA consists of two parts - a generic Linux component with RTNETLINK callbacks to listen for multicast callbacks and a proprietary compile time driver to interface with a proprietary accelerator.

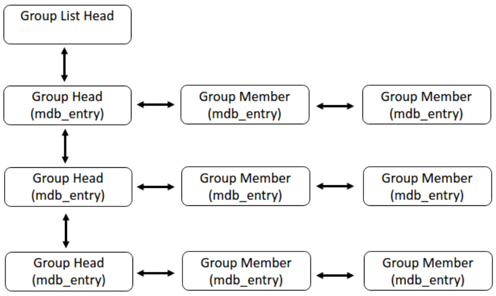
The RTNETLINK callbacks consist of two components - bridge MDB callbacks and router ROUTE callbacks. The bridge MDB callbacks consist of NEWMDB and DELMDB messages that result in join and leave operations respectively. The ROUTE callbacks consist of NEWROUTE and DELROUTE messages of type RTN_MULTICAST.

The following diagram illustrates the basic operation of the model.



Internally the logic of MCAST-PA is similar to the multicast component of the Linux bridge. Essentially MCAST-PA is a list manager, maintaining a list of multicast groups. For each multicast group there is a list of group members. When a NEWMDB message is received for a new group, a group head and group member are created. When a multicast route is received for that group, the group member(s) are pushed to the accelerator via pa_join(). When a NEWMDB message is received for an existing route the pa_join() is executed immediately.

The following diagram illustrates the list management component.



It should be noted that under this model the Linux bridge will software bridge a few packets until the entry is pushed to the accelerator. This has a beneficial affect of minimizing the first packet delay to the STB.

Since the Linux bridge supports multicast to unicast conversion, the bridge also tracks source MAC addresses of the STBs. To support source MAC requirements of the Intel PA, MCAST-PA also needs to maintain source MAC addresses within its internal MDB Entry structure (from if_bridge.h).

To support this a patch was added to the Linux bridge to pass source MAC address in the NEWMDB and DELMDB messages. There is also a requirement for Network Operators to know what devices are watching various channels. Both requirements can be addressed via MDB Entry and bridge patches/enhancements. There is also a patch that forces the bridge to track source MAC addresses even when multicast to unicast conversion is not enabled.

1.1.1 Invocation

Three types of video delivery are supported: routed from a single wan interface(proxy), routed from a dedicated video interface(proxy) and pure bridge with a dedicated video interface attached to a dedicated video bridge(snoop).

For the single wan interface case, MCAST-PA just needs to know the wan interface:

```
mcast-pa --wan <wan interface name>
```

For the dedicate video wan interface case, MCAST-PA needs to know the wan interface, the bridge name and the "video2lan" mode. Note that video clients are still attached to br-lan but the IP address is held by the video bridge with the video wan interface a member of the bridge.

```
mcast-pa --wan <wan interface name> --video2lan <video bridge device name>
```

For the pure bridge case, MCAST-PA needs to know the wan interface and the bridge name:

```
mcast-pa --wan <wan interface name> --bridge <video bridge device name>
```

1.1.2 Logging

By default syslog LOG_NOTICE is turned on so channel changes can be monitored:

```
Tue Jul 17 07:40:30 2018 local1.notice mcast-pa[13135]: pa_leave:401 leave group 224.0.18.113 wan wan lan wifi
Tue Jul 17 07:40:37 2018 local1.notice mcast-pa[13135]: RTM_NEWMDB dev br-lan port wifi5g grp 224.0.18.115 src
Tue Jul 17 07:40:37 2018 local1.notice mcast-pa[13135]: pa_join:363 join new group 224.0.18.115 wan wan lan wi
Tue Jul 17 07:40:40 2018 local1.notice mcast-pa[13135]: RTM_DELMDB dev br-lan port wifi5g grp 224.0.18.114 src
Tue Jul 17 07:40:40 2018 local1.notice mcast-pa[13135]: pa_leave:401 leave group 224.0.18.114 wan wan lan wifi
Tue Jul 17 07:40:47 2018 local1.notice mcast-pa[13135]: RTM_NEWMDB dev br-lan port wifi5g grp 224.0.18.116 src
Tue Jul 17 07:40:47 2018 local1.notice mcast-pa[13135]: pa_join:363 join new group 224.0.18.116 wan wan lan wi
```

With the optional command line arguement –verbose, all netlink messages are also logged with a LOG_INFO.

1.1.3 MCPROXY

As mentioned above the basic model independent of Packet Acceleration is MCPROXY with a Linux bridge. The MCPROXY package can be found here:

```
https://github.com/openwrt-routing/packages/tree/master/mcproxy
```

A sample MCPROXY configuration for the simple single wan case:

```
/etc/config/mcproxy:

config mcproxy ’mcproxy’
      option disabled ’0’
      option respawn ’1’
      option protocol ’IGMPv2’

config instance
      option name ’wan2lan’
      option upstream ’wan’
      option downstream ’br-lan’
      option disabled ’0’
```

```
/var/etc/mcproxy_mcproxy.conf:

protocol IGMPv2;

pinstance wan2lan: "wan" ==> "br-lan";
```

Adding a source address structure to the above would meet both requirements as follows:

```
struct br_mdb_entry {
    __u32 ifindex;
#define MDB_TEMPORARY 0
#define MDB_PERMANENT 1
    __u8 state;
    struct {
        union {
            __be32  ip4;
            struct in6_addr ip6;
        } u;
        __be16      proto;
    } addr;
    struct {
        union {
            __be32  ip4;
            struct in6_addr ip6;
        } u;
        __be16      proto;
    unsigned char eth_addr[ETH_ALEN];
    } src_addr;
};
```

Chapter 2

Todo List

Global mcg_br_entry_join (struct mcg_br_mdb_entry_t ∗head)

IPV6

Note

Author

tim.hayes@smartrg.com

Global mcg_br_entry_srcmac_set (struct mcg_br_mdb_entry_t ∗mcge, struct mcastpa_join_leave_t ∗mjl)

someday get a really clean way of doing this

Note

done to support Intel wifi mc which requires source of video subscriber

Author

tim.hayes@smartrg.com

Global pa_deinit (struct mcastpa_system_init_t ∗msi)

flush entries

Author

tim.hayes@smartrg.com

Global parse_br_mdb_entry (struct nlmsghdr ∗n, int ifindex, struct rtattr ∗attr)

we only look for one wan interface.

what about guest bridge - need to check for that

Author

tim.hayes@smartrg.com

Global parse_route (const struct sockaddr_nl ∗who, struct nlmsghdr ∗n, void ∗arg)

support for multihoming and review

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Data Structure Documentation

5.1 mcast_ip_entry_t Struct Reference

Data Fields

- struct list_head head
- int ifindex
- char address [INET_ADDR_SIZE]
- char name [IFNAMSIZ]

5.1.1 Detailed Description

Definition at line 264 of file mcast-pa.c.

5.1.2 Field Documentation

5.1.2.1 char mcast_ip_entry_t::address[INET_ADDR_SIZE]

ip address of device

Definition at line 267 of file mcast-pa.c.

Referenced by mcast_ip_entry_add(), mcast_ip_entry_del(), mcast_ip_entry_get(), and mcast_ip_entry_list().

5.1.2.2 struct list_head mcast_ip_entry_t::head

prev next pointers for ip address list

Definition at line 265 of file mcast-pa.c.

Referenced by mcast_ip_entry_add(), mcast_ip_entry_del(), mcast_ip_entry_get(), and mcast_ip_entry_list().

5.1.2.3 int mcast_ip_entry_t::ifindex

ifindex of device which has address

Definition at line 266 of file mcast-pa.c.

5.1.2.4 char mcast_ip_entry_t::name[IFNAMSIZ]

name of device

Definition at line 268 of file mcast-pa.c.

The documentation for this struct was generated from the following file:

- mcast-pa.c

5.2 mcast_wan_entry_t Struct Reference

Data Fields

- struct list_head head
- int ifindex
- char address [INET_ADDR_SIZE]
- char name [IFNAMSIZ]

5.2.1 Detailed Description

Definition at line 271 of file mcast-pa.c.

5.2.2 Field Documentation

5.2.2.1 char mcast_wan_entry_t::address[INET_ADDR_SIZE]

ip address of device

Definition at line 274 of file mcast-pa.c.

5.2.2.2 struct list_head mcast_wan_entry_t::head

prev next pointers for ip address list

Definition at line 272 of file mcast-pa.c.

Referenced by mcast_wan_entry_add(), mcast_wan_entry_default(), mcast_wan_entry_del(), mcast_wan_entry_get(), and mcast_wan_entry_list().

5.2.2.3 int mcast_wan_entry_t::ifindex

ifindex of device which has address

Definition at line 273 of file mcast-pa.c.

5.2.2.4 char mcast_wan_entry_t::name[IFNAMSIZ]

name of device

Definition at line 275 of file mcast-pa.c.

Referenced by mcast_wan_entry_add(), mcast_wan_entry_default(), mcast_wan_entry_del(), mcast_wan_entry_get(), and mcast_wan_entry_list().

The documentation for this struct was generated from the following file:

- mcast-pa.c

5.3 mcastpa_join_leave_t Struct Reference

```
#include <mcast-pa.h>
```

Data Fields

- int flags
- char group [MCASTPA_STRING_SIZE]
- char srcip [MCASTPA_STRING_SIZE]
- char wan [MCASTPA_STRING_SIZE]
- char lan [MCASTPA_STRING_SIZE]
- char lan_dev [MCASTPA_STRING_SIZE]
- char srcmac [ETH_ALEN]

5.3.1 Detailed Description

Definition at line 40 of file mcast-pa.h.

5.3.2 Field Documentation

5.3.2.1 int mcastpa_join_leave_t::flags

bridge, srcip valid etc.

Definition at line 47 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), mcg_br_entry_leave(), and pa_join().

5.3.2.2 char mcastpa_join_leave_t::group[MCASTPA_STRING_SIZE]

ascii string of ip mc group e.g. 224.0.18.101

Definition at line 48 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), mcg_br_entry_leave(), pa_join(), and pa_leave().

5.3.2.3 char mcastpa_join_leave_t::lan[MCASTPA_STRING_SIZE]

ascii string names of lan interfaces e.g. lan1 lan2 wifi5g etc

Definition at line 51 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), and mcg_br_entry_leave().

5.3.2.4 char mcastpa_join_leave_t::lan_dev[MCASTPA_STRING_SIZE]

ascii string names of lan interfaces that is joined or leaved

Definition at line 52 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), mcg_br_entry_leave(), pa_join(), and pa_leave().

5.3.2.5 char mcastpa_join_leave_t::srcip[MCASTPA_STRING_SIZE]

ascii string name of video source ip

Definition at line 49 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), mcg_br_entry_leave(), pa_join(), and pa_leave().

5.3.2.6 char mcastpa_join_leave_t::srcmac[ETH_ALEN]

source mac address of group subscriber

Definition at line 53 of file mcast-pa.h.

Referenced by mcg_br_entry_srcmac_set(), pa_join(), and pa_leave().

5.3.2.7 char mcastpa_join_leave_t::wan[MCASTPA_STRING_SIZE]

ascii string name of wan video ingress device

Definition at line 50 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), mcg_br_entry_leave(), pa_join(), and pa_leave().

The documentation for this struct was generated from the following file:

- mcast-pa.h

5.4 mcastpa_system_init_t Struct Reference

```
#include <mcast-pa.h>
```

Data Fields

- int flags
- char srcip [MCASTPA_STRING_SIZE]
- char wan [MCASTPA_STRING_SIZE]

5.4.1 Detailed Description

Definition at line 33 of file mcast-pa.h.

5.4.2 Field Documentation

5.4.2.1 int mcastpa_system_init_t::flags

bridge, srcip valid etc.

Definition at line 35 of file mcast-pa.h.

5.4.2.2 char mcastpa_system_init_t::srcip[MCASTPA_STRING_SIZE]

ascii string name of video source ip

Definition at line 36 of file mcast-pa.h.

5.4.2.3 char mcastpa_system_init_t::wan[MCASTPA_STRING_SIZE]

ascii string name of wan video ingress device

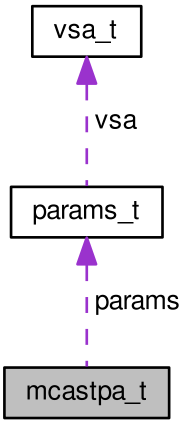
Definition at line 37 of file mcast-pa.h.

The documentation for this struct was generated from the following file:

- mcast-pa.h

5.5 mcastpa_t Struct Reference

Collaboration diagram for mcastpa_t:



Data Fields

- struct params_t params
- uint64_t timer_tick
- struct ip_mreq group
- int sock
- char dev_name [IFNAMSIZ]
- struct timespec current_time
- struct timespec last_time
- struct timespec start_time
- time_t start_ctime
- time_t error_ctime
- struct list_head mcg_head
- struct list_head ip_head
- struct list_head wan_head

5.5.1 Detailed Description

Definition at line 313 of file mcast-pa.c.

5.5.2 Field Documentation

5.5.2.1 struct timespec mcastpa_t::current_time

current time from timer handler

Definition at line 319 of file mcast-pa.c.

5.5.2.2 char mcastpa_t::dev_name[IFNAMSIZ]

video ingress device name

Definition at line 318 of file mcast-pa.c.

5.5.2.3 time_t mcastpa_t::error_ctime

error detected time in local time format when we started

Definition at line 323 of file mcast-pa.c.

5.5.2.4 struct ip_mreq mcastpa_t::group

group struct used for joins

Definition at line 316 of file mcast-pa.c.

5.5.2.5 struct list_head mcastpa_t::ip_head

global list header for our host ip addresses

Definition at line 325 of file mcast-pa.c.

Referenced by main(), mcast_ip_entry_add(), mcast_ip_entry_del(), mcast_ip_entry_get(), and mcast_ip_entry_list().

5.5.2.6 struct timespec mcastpa_t::last_time

last time in timer handler - used to get actual interval - about 1 second plus or minus

Definition at line 320 of file mcast-pa.c.

5.5.2.7 struct list_head mcastpa_t::mcg_head

global list header for mc groups

Definition at line 324 of file mcast-pa.c.

Referenced by main(), mcg_br_entry_head_add(), mcg_br_entry_head_del(), mcg_br_entry_head_del_all(), mcg_br_entry_head_get(), mcg_br_entry_head_get_from_group(), mcg_br_entry_head_list_del_all(), and mcg_br_entry_head_list_show().

5.5.2.8 struct params_t mcastpa_t::params

command line args on invocation

Definition at line 314 of file mcast-pa.c.

Referenced by cache_mdb_entry(), do_monitor(), do_wait_wan(), iswan(), main(), mcast_sig_handler(), mcast_vsa_get(), mcg_br_entry_join(), mcg_br_entry_leave(), parse_br_mdb_entry(), vsa_entry_join(), vsa_entry_leave(), vsa_entry_process(), and vsa_parse_init().

5.5.2.9 int mcastpa_t::sock

socket used for joins

Definition at line 317 of file mcast-pa.c.

5.5.2.10 time_t mcastpa_t::start_ctime

start time in local time format when we started

Definition at line 322 of file mcast-pa.c.

5.5.2.11 struct timespec mcastpa_t::start_time

time that we started

Definition at line 321 of file mcast-pa.c.

5.5.2.12 uint64_t mcastpa_t::timer_tick

increment each 1 second timer tick in timer handler

Definition at line 315 of file mcast-pa.c.

5.5.2.13 struct list_head mcastpa_t::wan_head

global list header for our host interfaces

Definition at line 326 of file mcast-pa.c.

Referenced by main(), mcast_wan_entry_add(), mcast_wan_entry_default(), mcast_wan_entry_del(), mcast_wan_entry_get(), and mcast_wan_entry_list().

The documentation for this struct was generated from the following file:

- mcast-pa.c

5.6 mcg_br_mdb_entry_t Struct Reference

Data Fields

- struct list_head mcg_head
- struct list_head mcg_entry
- struct br_mdb_entry e
- int joined
- int wan_ifindex
- int br_ifindex
- char src [INET_ADDR_SIZE]

5.6.1 Detailed Description

Definition at line 278 of file mcast-pa.c.

5.6.2 Field Documentation

5.6.2.1 int mcg_br_mdb_entry_t::br_ifindex

ifindex of bridge interface - head use only

Definition at line 284 of file mcast-pa.c.

Referenced by cache_mdb_entry(), mcg_br_entry_head_show(), mcg_br_entry_show(), and vsa_entry_join().

5.6.2.2 struct br_mdb_entry mcg_br_mdb_entry_t::e

copy of mdb entry from bridge table with mc group and ifindex of joined interfaces

Definition at line 281 of file mcast-pa.c.

Referenced by mcg_br_entry_add(), mcg_br_entry_del(), mcg_br_entry_get(), mcg_br_entry_head_add(), mcg_br_entry_head_del(), mcg_br_entry_head_del_all(), mcg_br_entry_head_get(), mcg_br_entry_head_get_from_group(), mcg_br_entry_head_show(), mcg_br_entry_join(), mcg_br_entry_leave(), mcg_br_entry_show(), and mcg_br_entry_srcmac_set().

5.6.2.3 int mcg_br_mdb_entry_t::joined

set to 1 if pa_join() called

Definition at line 282 of file mcast-pa.c.

Referenced by mcg_br_entry_join(), and mcg_br_entry_leave().

5.6.2.4 struct list_head mcg_br_mdb_entry_t::mcg_entry

prev next pointers for mc group interface members (list of ifindexes via br_mdb_entry struct - head use only

Definition at line 280 of file mcast-pa.c.

Referenced by cache_mdb_entry(), mcg_br_entry_add(), mcg_br_entry_del(), mcg_br_entry_get(), mcg_br_entry_head_add(), mcg_br_entry_join(), mcg_br_entry_leave(), mcg_br_entry_list_show(), and vsa_entry_leave().

5.6.2.5 struct list_head mcg_br_mdb_entry_t::mcg_head

prev next pointers for mc group header list mgmt (list of groups)

Definition at line 279 of file mcast-pa.c.

Referenced by mcg_br_entry_head_add(), mcg_br_entry_head_del(), mcg_br_entry_head_del_all(), mcg_br_entry_head_get(), mcg_br_entry_head_get_from_group(), mcg_br_entry_head_list_del_all(), and mcg_br_entry_head_list_show().

5.6.2.6 char mcg_br_mdb_entry_t::src[INET_ADDR_SIZE]

ip address of video source - head use only

Definition at line 285 of file mcast-pa.c.

Referenced by do_mroute(), mcg_br_entry_head_show(), mcg_br_entry_join(), and mcg_br_entry_leave().

5.6.2.7 int mcg_br_mdb_entry_t::wan_ifindex

ifindex of wan interface - head use only

Definition at line 283 of file mcast-pa.c.

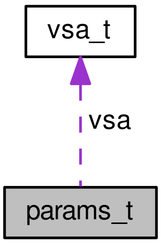
Referenced by do_mroute(), and mcg_br_entry_head_show().

The documentation for this struct was generated from the following file:

- mcast-pa.c

5.7 params_t Struct Reference

Collaboration diagram for params_t:



Data Fields

- int dbg
- int foreground
- int verbose
- int monitor
- int wan
- int wan_ifindex
- int bridged
- char bridge_name [IFNAMSIZ]
- int video2lan
- char video2lan_name [IFNAMSIZ]
- int use_src
- int nowifi
- char src [INET_ADDR_SIZE]
- int exp
- struct vsa_t vsa

5.7.1 Detailed Description

Definition at line 295 of file mcast-pa.c.

5.7.2 Field Documentation

5.7.2.1 char params_t::bridge_name[IFNAMSIZ]

name of video bridge e.g. br-lan or br-video

Definition at line 303 of file mcast-pa.c.

Referenced by main(), and parse_br_mdb_entry().

5.7.2.2 int params_t::bridged

set if in bridge mode

Definition at line 302 of file mcast-pa.c.

Referenced by do_monitor(), main(), mcg_br_entry_join(), mcg_br_entry_leave(), and parse_br_mdb_entry().

5.7.2.3 int params_t::dbg

set if debug output is desired

Definition at line 296 of file mcast-pa.c.

Referenced by main().

5.7.2.4 int params_t::exp

experimental code segment testing

Definition at line 309 of file mcast-pa.c.

Referenced by main().

5.7.2.5 int params_t::foreground

set if we are to run in foreground - background daemon is default

Definition at line 297 of file mcast-pa.c.

Referenced by main().

5.7.2.6 int params_t::monitor

set to monitor NEWMDB and DELMDB

Definition at line 299 of file mcast-pa.c.

Referenced by main().

5.7.2.7 int params_t::nowifi

don’t push wifi ifaces to packet accelerator

Definition at line 307 of file mcast-pa.c.

Referenced by cache_mdb_entry(), and main().

5.7.2.8 char params_t::src[INET_ADDR_SIZE]

ip address of video source from command line

Definition at line 308 of file mcast-pa.c.

Referenced by main(), mcg_br_entry_join(), and mcg_br_entry_leave().

5.7.2.9 int params_t::use_src

use ip address of video source from command line

Definition at line 306 of file mcast-pa.c.

Referenced by main(), mcg_br_entry_join(), and mcg_br_entry_leave().

5.7.2.10 int params_t::verbose

set to generate verbose output

Definition at line 298 of file mcast-pa.c.

Referenced by main().

5.7.2.11 int params_t::video2lan

set if in video2lan is like bridge mode but don't care which bridge

Definition at line 304 of file mcast-pa.c.

Referenced by iswan(), and main().

5.7.2.12 char params_t::video2lan_name[IFNAMSIZ]

name of video ip interface

Definition at line 305 of file mcast-pa.c.

Referenced by iswan(), and main().

5.7.2.13 struct vsa_t params_t::vsa

for vsa join and leave operations

Definition at line 310 of file mcast-pa.c.

Referenced by mcast_sig_handler(), mcast_vsa_get(), vsa_entry_join(), vsa_entry_leave(), vsa_entry_process(), and vsa_parse_init().

5.7.2.14 int params_t::wan

set if wan input provided

Definition at line 300 of file mcast-pa.c.

Referenced by main().

5.7.2.15 int params_t::wan_ifindex

set if wan exists on boot

Definition at line 301 of file mcast-pa.c.

Referenced by do_wait_wan(), and parse_br_mdb_entry().

The documentation for this struct was generated from the following file:

- mcast-pa.c

5.8 vsa_t Struct Reference

Data Fields

- char op [128]
- char group [128]
- char device [128]
- int valid

5.8.1 Detailed Description

Definition at line 288 of file mcast-pa.c.

5.8.2 Field Documentation

5.8.2.1 char vsa_t::device[128]

vsa device - like a lan bridge port

Definition at line 291 of file mcast-pa.c.

Referenced by mcast_vsa_get(), vsa_entry_join(), and vsa_entry_leave().

5.8.2.2 char vsa_t::group[128]

vsa mc group

Definition at line 290 of file mcast-pa.c.

Referenced by mcast_vsa_get(), vsa_entry_join(), and vsa_entry_leave().

5.8.2.3 char vsa_t::op[128]

vsa join or leave

Definition at line 289 of file mcast-pa.c.

Referenced by mcast_vsa_get(), and vsa_entry_process().

5.8.2.4 int vsa_t::valid

valid data in this struct

Definition at line 292 of file mcast-pa.c.

Referenced by mcast_sig_handler(), mcast_vsa_get(), and vsa_parse_init().

The documentation for this struct was generated from the following file:

- mcast-pa.c

Chapter 6

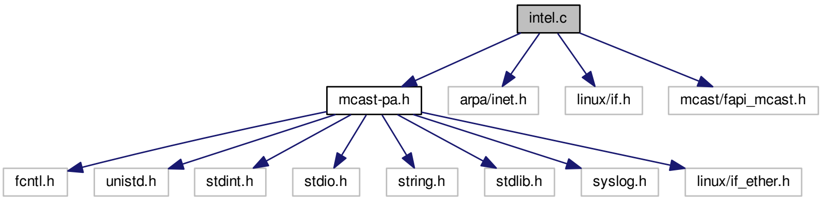
File Documentation

6.1 intel.c File Reference

Intel Multicast Packet Accelerator.

```
#include <mcast-pa.h>
#include <arpa/inet.h>
#include <linux/if.h>
#include <mcast/fapi_mcast.h>
```

Include dependency graph for intel.c:



Macros

- #define INTEL_MCAST_USE_MCAST_FAPI 1
- #define MCAST_HELPER_DEV_MAJOR_NUM 240
- #define MCAST_HELPER_DEVICE "/dev/mcast"
- #define MCAST_HELPER_DEV_MINOR_NUM 0

Functions

- int fapi_mch_init_sos (void)

  inits intel mcast_helper module
- int pa_init_sos (struct mcastpa_system_init_t ∗msi)

  inits intel mcast fapi subsystem
- int pa_init (struct mcastpa_system_init_t ∗msi)

  inits intel mcast fapi subsystem using fapi_mcast.c (libmcastfapi)
- int pa_join (struct mcastpa_join_leave_t ∗mjl)

  joins by libmcastfapi call

- int pa_leave (struct mcastpa_join_leave_t ∗mjl)

  leaves by libmcastfapi call
- int pa_deinit (struct mcastpa_system_init_t ∗msi)

  de-inits intel mcast subsystem

6.1.1 Detailed Description

Intel Multicast Packet Accelerator.

Author

tim.hayes@smartrg.com

Date

Spring 2016

Adds and deletes multicast groups to Intel PPA subsystem using ppacmd, mcast_cli or libmcastfapi

Definition in file intel.c.

6.1.2 Macro Definition Documentation

6.1.2.1 #define INTEL_MCAST_USE_MCAST_FAPI 1

Definition at line 38 of file intel.c.

6.1.2.2 #define MCAST_HELPER_DEV_MAJOR_NUM 240

Definition at line 257 of file intel.c.

Referenced by fapi_mch_init_sos().

6.1.2.3 #define MCAST_HELPER_DEV_MINOR_NUM 0

Definition at line 259 of file intel.c.

Referenced by fapi_mch_init_sos().

6.1.2.4 #define MCAST_HELPER_DEVICE "/dev/mcast"

Definition at line 258 of file intel.c.

Referenced by fapi_mch_init_sos().

6.1.3 Function Documentation

6.1.3.1 int fapi_mch_init_sos (void)

inits intel mcast_helper module

from fapi_mcast.c (libmcastfapi) - a subset without igmp, iptables init ...

Note

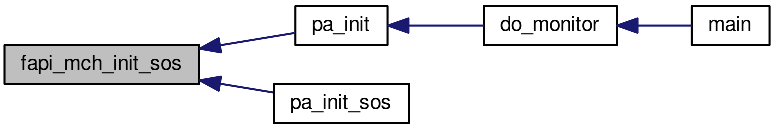
Author

tim.hayes@smartrg.com

Definition at line 270 of file intel.c.

Referenced by pa_init(), and pa_init_sos().

Here is the caller graph for this function:



6.1.3.2 int pa_deinit (struct mcastpa_system_init_t * *msi*)

de-inits intel mcast subsystem

not used because it unload the module

Note

Todo flush entries

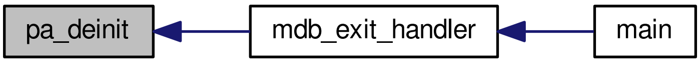
Author

tim.hayes@smartrg.com

Definition at line 416 of file intel.c.

Referenced by mdb_exit_handler().

Here is the caller graph for this function:



6.1.3.3 int pa_init (struct mcastpa_system_init_t * *msi*)

inits intel mcast fapi subsystem using fapi_mcast.c (libmcastfapi)

Note

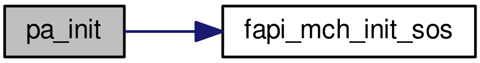
Author

tim.hayes@smartrg.com

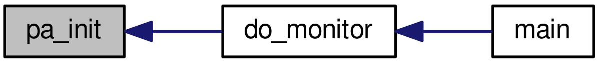
Definition at line 317 of file intel.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.4 int pa_init_sos (struct mcastpa_system_init_t * *msi*)

inits intel mcast fapi subsystem

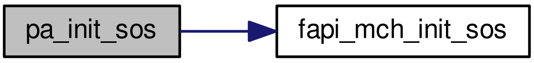
Note

Author

tim.hayes@smartrg.com

Definition at line 300 of file intel.c.

Here is the call graph for this function:



6.1.3.5 int pa_join (struct mcastpa_join_leave_t ∗ *mjl*)

joins by libmcastfapi call

first group join is add additions are updates

Note

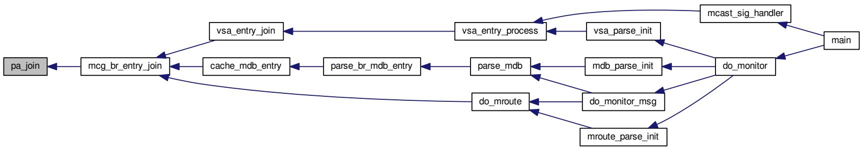
Author

tim.hayes@smartrg.com

Definition at line 340 of file intel.c.

Referenced by mcg_br_entry_join().

Here is the caller graph for this function:



6.1.3.6 int pa_leave (struct mcastpa_join_leave_t ∗ *mjl*)

leaves by libmcastfapi call

all group leaves are deletes no matter how many members

Note

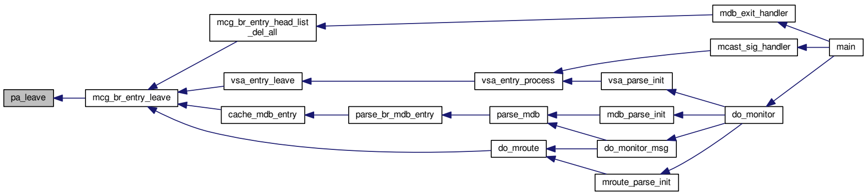
Author

tim.hayes@smartrg.com

Definition at line 387 of file intel.c.

Referenced by mcg_br_entry_leave().

Here is the caller graph for this function:

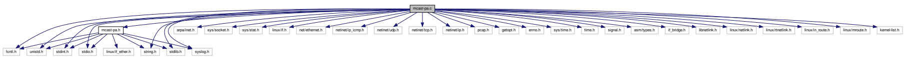


6.2 mcast-pa.c File Reference

Multicast Packet Accelerator.

```
#include <fcntl.h>
#include <unistd.h>
#include <stdint.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <syslog.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <linux/if.h>
#include <net/ethernet.h>
#include <netinet/ip_icmp.h>
#include <netinet/udp.h>
#include <netinet/tcp.h>
#include <netinet/ip.h>
#include <pcap.h>
#include <getopt.h>
#include <errno.h>
#include <sys/time.h>
#include <time.h>
#include <signal.h>
#include <asm/types.h>
#include "if_bridge.h"
#include <libnetlink.h>
#include <linux/netlink.h>
#include <linux/rtnetlink.h>
#include <linux/in_route.h>
#include <linux/mroute.h>
#include <kernel-list.h>
#include <mcast-pa.h>
```

Include dependency graph for mcast-pa.c:



Data Structures

- struct mcast_ip_entry_t
- struct mcast_wan_entry_t
- struct mcg_br_mdb_entry_t
- struct vsa_t
- struct params_t
- struct mcastpa_t

Macros

- #define INET_ADDR_SIZE 128
- #define CMD_BUF_SIZE 256
- #define SPRINT_BSIZE 64
- #define SPRINT_BUF(x) static char x[SPRINT_BSIZE]
- #define MDBA_RTA(r) ((struct rtattr∗)(((char∗)(r)) + NLMSG_ALIGN(sizeof(struct br_port_msg))))
- #define MDB_FORK_EXIT 69

  exit handler
- #define PRTFL(fl, flname)
- #define MROUTE_RATE_LIMIT_ENDLESS 0

  sets br-video to be a virtual router port
- #define MROUTE_TTL_THRESHOLD 1
- #define MROUTE_DEFAULT_TTL 1

Functions

- const char ∗ ll_index_to_name (unsigned idx)
- unsigned ll_name_to_index (const char ∗name)
- void ll_init_map (struct rtnl_handle ∗rth)
- int mcg_br_entry_leave (struct mcg_br_mdb_entry_t ∗head, struct br_mdb_entry ∗e)

  leaves with no groups or joins with smaller group list
- static __u32 nl_mgrp (__u32 group)
- int mcast_vsa_get (void)

  read a file and look for a vsa operation
- struct mcast_wan_entry_t ∗ mcast_wan_entry_add (char ∗name)

  add a wan iface to our host list
- int mcast_wan_entry_del (char ∗name)

  delete a wan iface if found on list
- struct mcast_wan_entry_t ∗ mcast_wan_entry_get (char ∗name)

  returns a list entry of the wan iface
- char ∗ mcast_wan_entry_default (void)

  returns a first entry of the wan iface
- int mcast_wan_entry_list (void)

  list all host wan ifaces
- int iswan (char ∗name)

determines if a interface is a wan interface

- int iswifi (char ∗name)

  determines if a interface is a wifi interface
- struct mcast_ip_entry_t ∗ mcast_ip_entry_add (char ∗address)

  add a ip address to our host list
- int mcast_ip_entry_del (char ∗address)

  delete a ip address if found on list
- struct mcast_ip_entry_t ∗ mcast_ip_entry_get (char ∗address)

  returns a list entry of the ip address
- int islocaladdr (char ∗address)

  determines if an address is one of our own local
- int mcast_ip_entry_list (void)

  list all host ip addresses
- void mcg_br_entry_head_show (FILE ∗f, struct mcg_br_mdb_entry_t ∗head)

  show an instance of a mc group entry head
- void mcg_br_entry_show (FILE ∗f, struct mcg_br_mdb_entry_t ∗mcge)

  show an instance of a mc group entry
- int mcg_br_entry_equal (struct br_mdb_entry ∗ex, struct br_mdb_entry ∗ey)

  compare bridge entries
- struct mcg_br_mdb_entry_t ∗ mcg_br_entry_head_get (struct br_mdb_entry ∗e)

  gets a head instance of mc group
- struct mcg_br_mdb_entry_t ∗ mcg_br_entry_get (struct mcg_br_mdb_entry_t ∗head, struct br_mdb_entry ∗e)

  get a mdb entry from the mc group list head list of attached group entries
- struct mcg_br_mdb_entry_t ∗ mcg_br_entry_add (struct mcg_br_mdb_entry_t ∗head, struct br_mdb_entry ∗e)

  add a bridge mdb entry to the mc group list head list
- struct mcg_br_mdb_entry_t ∗ mcg_br_entry_head_add (struct br_mdb_entry ∗e)

  add a bridge mdb entry to the mc group list head
- int mcg_br_entry_head_del (struct mcg_br_mdb_entry_t ∗head)

  delete a bridge mdb entry from the mc group list head
- int mcg_br_entry_del (struct mcg_br_mdb_entry_t ∗head, struct br_mdb_entry ∗e)

  delete a bridge mdb entry from the mc group list head list
- void mcg_br_entry_head_del_all (struct mcg_br_mdb_entry_t ∗head)

  delete a bridge mdb entry from the mc group list head
- int mcg_br_entry_head_list_del_all (void)

  traverses all head groups, removes all sub entries issues leave and deletes head
- void mcg_br_entry_list_show (FILE ∗f, struct mcg_br_mdb_entry_t ∗head)

  lists instances of a mc group entires tied to a mc group head entry
- void mcg_br_entry_srcmac_set (struct mcg_br_mdb_entry_t ∗mcge, struct mcastpa_join_leave_t ∗mjl)

  sets src mac in mjl struct
- int mcg_br_entry_join (struct mcg_br_mdb_entry_t ∗head)

  joins a new group
- void mcg_br_entry_head_list_show (void)

  lists instances of head a mc group entires
- void vsa_entry_join (void)

  joins a vsa requested group on a device
- void vsa_entry_leave (void)

  leave a vsa requested group on a device
- void vsa_entry_process (void)

  process a vsa request

- struct mcg_br_mdb_entry_t ∗ mcg_br_entry_head_get_from_group (char ∗group)

  finds a head entry from a specific mc group
- static void mdb_exit_handler (int ev, void ∗arg)
- const char ∗ rt_addr_n2a (int af, const void ∗addr, char ∗buf, int buflen)

  misc parse_route support function
- char ∗ format_host (int af, int len, const void ∗addr, char ∗buf, int buflen)

  misc parse_route support function
- int af_bit_len (int af)

  misc parse_route support function
- static int rtm_get_table (struct rtmsg ∗r, struct rtattr ∗∗tb)

  misc parse_route support function
- static void print_rtax_features (FILE ∗fp, unsigned int features)

  misc parse_route support function
- int parse_route (const struct sockaddr_nl ∗who, struct nlmsghdr ∗n, void ∗arg)

  parses route update and gets our host ip address
- static int iproute_parse_init (void)

  initialize our wan route knowledge
- static char ∗ cache_mdb_entry_srcmac (struct br_mdb_entry ∗e)
- static void cache_mdb_entry (struct nlmsghdr ∗n, int ifindex, struct br_mdb_entry ∗e)
- static void parse_br_mdb_entry (struct nlmsghdr ∗n, int ifindex, struct rtattr ∗attr)

  parses mdb_entry - may contain mulitiples and calls cache function
- int parse_mdb (const struct sockaddr_nl ∗who, struct nlmsghdr ∗n, void ∗arg)
- int do_mroute (const struct sockaddr_nl ∗who, struct nlmsghdr ∗n, void ∗arg)
- static int do_monitor_msg (const struct sockaddr_nl ∗who, struct nlmsghdr ∗n, void ∗arg)
- static int mroute_parse_init (void)

  initialize our wan mroute knowledge
- void mroute_bridge_init (void)
- static int mdb_parse_init (void)

  initialize our lan mdb knowledge
- static int vsa_parse_init (void)

  handle any vsa requests
- int do_monitor (void)

  starts a netlink listener for routes, multicast routes and MDB changes
- int do_wait_wan (char ∗name)

  looks for wan device and sleep poll until it exists
- void mcast_sig_handler (int signo)

  sighandler
- void print_ip_header (unsigned char ∗Buffer, int Size)

  experimental code here
- void process_packet (unsigned char ∗buffer, int size)
- void do_exp (void)
- void mcastpa_usage (void)

  usage helper
- int main (int argc, char ∗∗argv)

  main entry point

Variables

- char ∗ _SL_ = "\n"
- static const char ∗ mx_names [RTAX_MAX+1]
- static struct mcastpa_t mcastpa
- struct rtnl_handle rth = {.fd = -1 }
- static struct option long_options [ ]

6.2.1 Detailed Description

Multicast Packet Accelerator.

Author

`tim.hayes@smartrg.com`

Date

Summer 2017

Adds and deletes multicast groups to PA (Intel PPA)

Definition in file mcast-pa.c.

6.2.2 Macro Definition Documentation

6.2.2.1 #define CMD_BUF_SIZE 256

Definition at line 230 of file mcast-pa.c.

6.2.2.2 #define INET_ADDR_SIZE 128

Definition at line 229 of file mcast-pa.c.

Referenced by mcg_br_entry_join(), mcg_br_entry_leave(), and parse_route().

6.2.2.3 #define MDB_FORK_EXIT 69

exit handler

Note

Author

`tim.hayes@smartrg.com`

Definition at line 1252 of file mcast-pa.c.

Referenced by main(), and mdb_exit_handler().

6.2.2.4 #define MDBA_RTA(*r*) ((struct rtattr∗)(((char∗)(r)) + NLMSG_ALIGN(sizeof(struct br_port_msg))))

Definition at line 256 of file mcast-pa.c.

Referenced by parse_mdb().

6.2.2.5 #define MROUTE_DEFAULT_TTL 1

Definition at line 2008 of file mcast-pa.c.

6.2.2.6 #define MROUTE_RATE_LIMIT_ENDLESS 0

sets br-video to be a virtual router port

Note

Author

tim.hayes@smartrg.com

Definition at line 2006 of file mcast-pa.c.

Referenced by do_exp(), and mroute_bridge_init().

6.2.2.7 #define MROUTE_TTL_THRESHOLD 1

Definition at line 2007 of file mcast-pa.c.

Referenced by do_exp(), and mroute_bridge_init().

6.2.2.8 #define PRTFL(*fl,* *flname*)

Value:

```
if (flags&RTCF_##fl) { \
  flags &= ~RTCF_##fl; \
  syslog (LOG_INFO,"%s" flname "%s", first ? "<" : "", flags ? "," : "> "); \
  first = 0; }
```

Referenced by parse_route().

6.2.2.9 #define SPRINT_BSIZE 64

Definition at line 252 of file mcast-pa.c.

6.2.2.10 #define SPRINT_BUF(*x*) static char x[SPRINT_BSIZE]

Definition at line 253 of file mcast-pa.c.

Referenced by cache_mdb_entry(), cache_mdb_entry_srcmac(), mcg_br_entry_head_get_from_group(), mcg_br-_entry_head_show(), mcg_br_entry_join(), mcg_br_entry_leave(), and mcg_br_entry_show().

6.2.3 Function Documentation

6.2.3.1 int af_bit_len (int *af*)

misc parse_route support function

Note

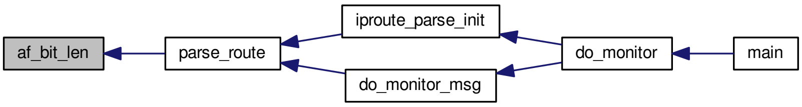
Author

tim.hayes@smartrg.com

Definition at line 1310 of file mcast-pa.c.

Referenced by parse_route().

Here is the caller graph for this function:

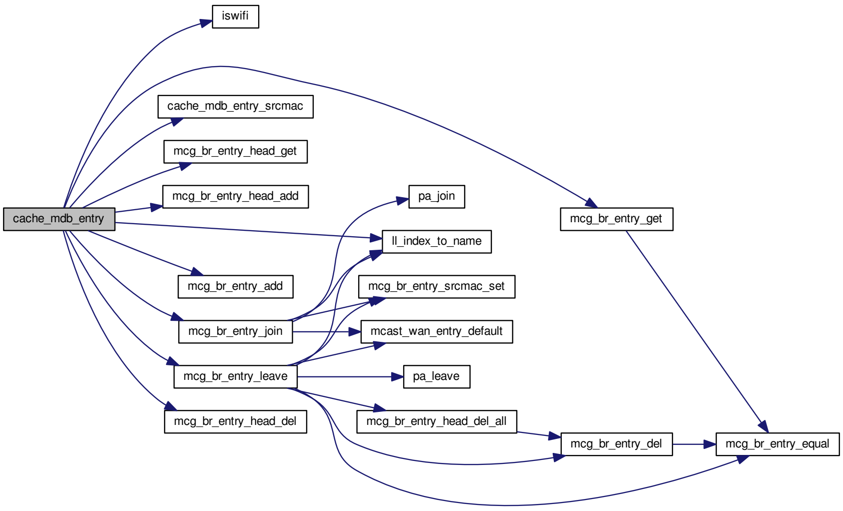


6.2.3.2 static void cache_mdb_entry (struct nlmsghdr ∗ *n,* int *ifindex,* struct br_mdb_entry ∗ *e*) [static]

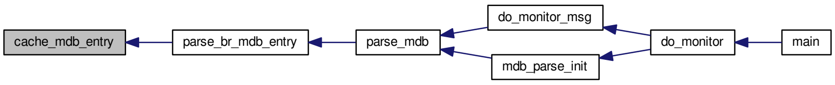
Definition at line 1684 of file mcast-pa.c.

Referenced by parse_br_mdb_entry().

Here is the call graph for this function:



Here is the caller graph for this function:

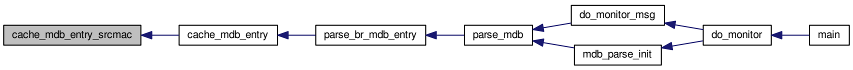


6.2.3.3 static char∗ cache_mdb_entry_srcmac (struct br_mdb_entry ∗ *e*) [inline], [static]

Definition at line 1668 of file mcast-pa.c.

Referenced by cache_mdb_entry().

Here is the caller graph for this function:

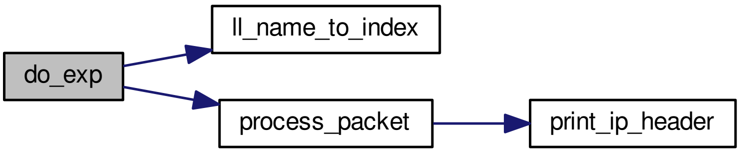


6.2.3.4 void do_exp (void)

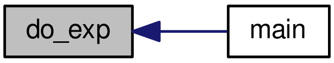
Definition at line 2277 of file mcast-pa.c.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.5 int do_monitor (void)

starts a netlink listener for routes, multicast routes and MDB changes

Note

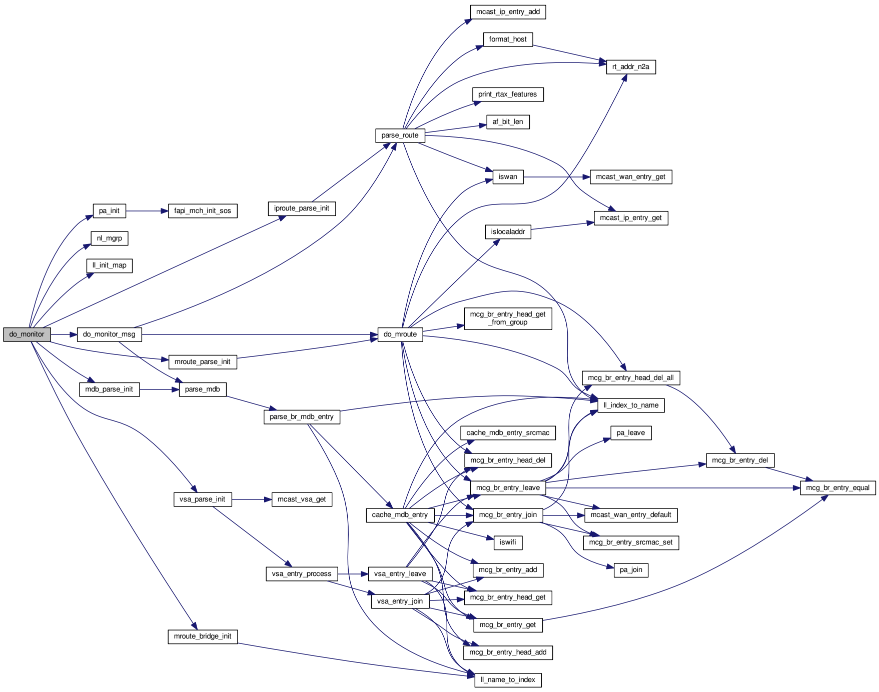
Author

tim.hayes@smartrg.com

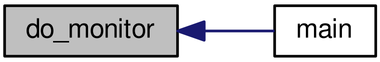
Definition at line 2099 of file mcast-pa.c.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

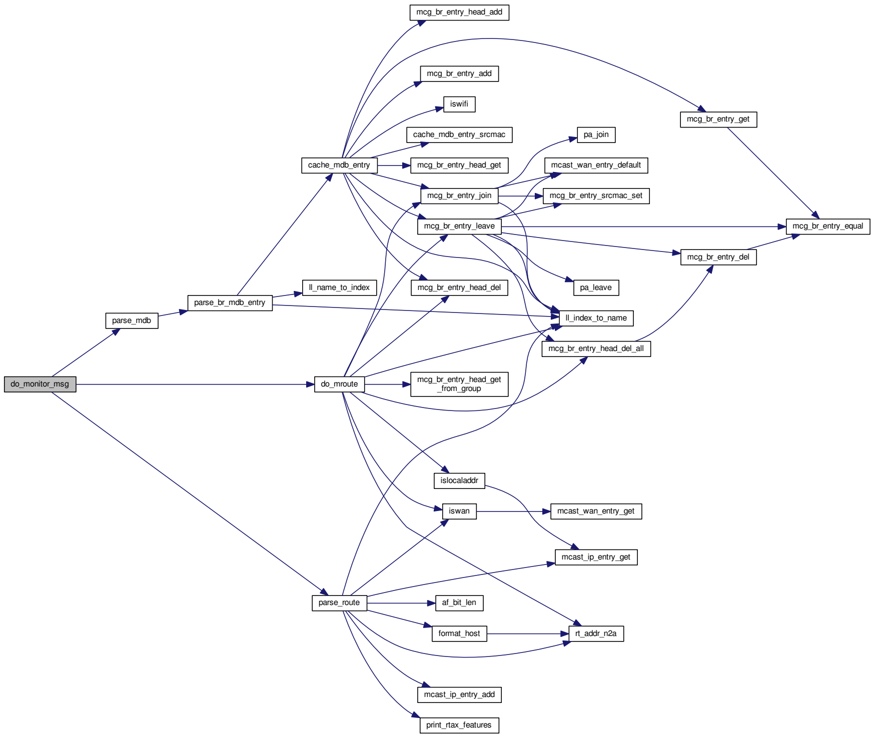


6.2.3.6 static int do_monitor_msg (const struct sockaddr_nl ∗ *who,* struct nlmsghdr ∗ *n,* void ∗ *arg*) [static]

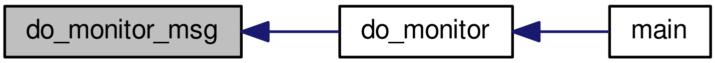
Definition at line 1938 of file mcast-pa.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:

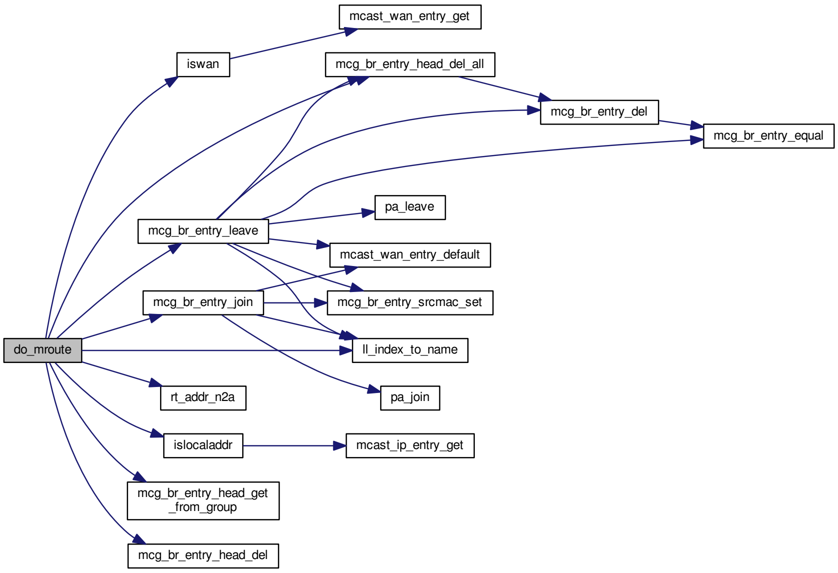


6.2.3.7 int do_mroute (const struct sockaddr_nl ∗ *who,* struct nlmsghdr ∗ *n,* void ∗ *arg*)

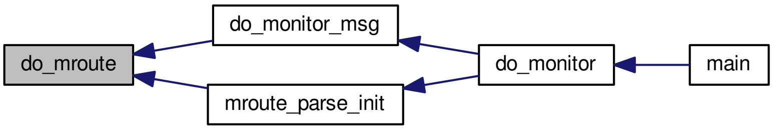
Definition at line 1834 of file mcast-pa.c.

Referenced by do_monitor_msg(), and mroute_parse_init().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.8 int do_wait_wan (char ∗ *name*)

looks for wan device and sleep poll until it exists

can be a startup race condition where we are started but netif hasn't created the device yet

Note

seen in Cspire CDT testing - never happen in default router mode

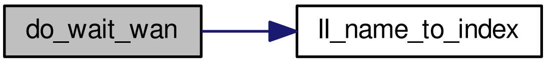
Author

tim.hayes@smartrg.com

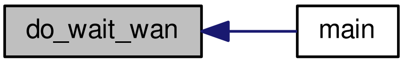
Definition at line 2158 of file mcast-pa.c.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.9 char∗ format_host (int *af,* int *len,* const void ∗ *addr,* char ∗ *buf,* int *buflen*)

misc parse_route support function

Note

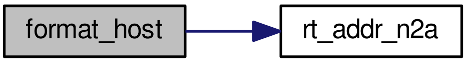
Author

tim.hayes@smartrg.com

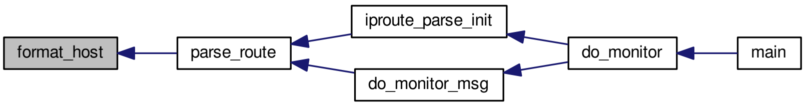
Definition at line 1296 of file mcast-pa.c.

Referenced by parse_route().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.10 static int iproute_parse_init (void) [static]

initialize our wan route knowledge

Note

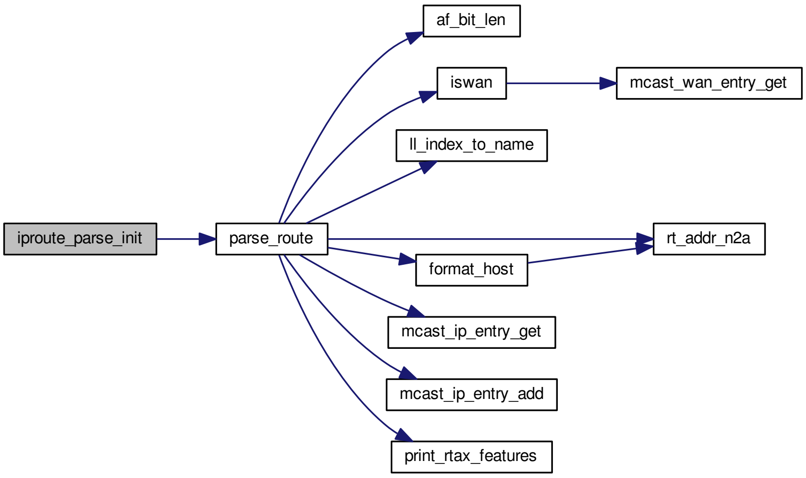
Author

tim.hayes@smartrg.com

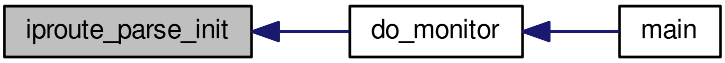
Definition at line 1652 of file mcast-pa.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.11 int islocaladdr (char ∗ *address*)

determines if an address is one of our own local

this is a temp hack need a list of adresses

Returns

1 if ours 0 otherwise

Note

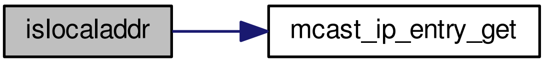
Author

tim.hayes@smartrg.com

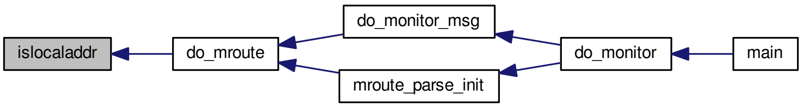
Definition at line 615 of file mcast-pa.c.

Referenced by do_mroute().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.12 int iswan (char ∗ *name*)

determines if a interface is a wan interface

Returns

1 if wan 0 otherwise

Note

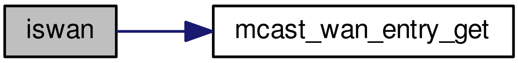
Author

tim.hayes@smartrg.com

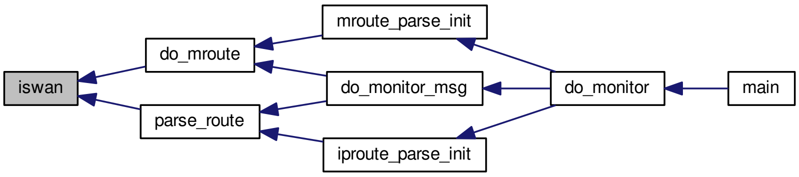
Definition at line 497 of file mcast-pa.c.

Referenced by do_mroute(), and parse_route().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.13 int iswifi (char ∗ *name*)

determines if a interface is a wifi interface

Returns

1 if wifi 0 otherwise

Note

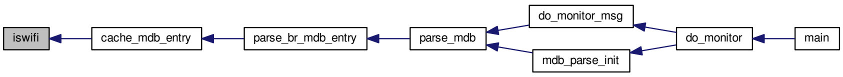
Author

tim.hayes@smartrg.com

Definition at line 523 of file mcast-pa.c.

Referenced by cache_mdb_entry().

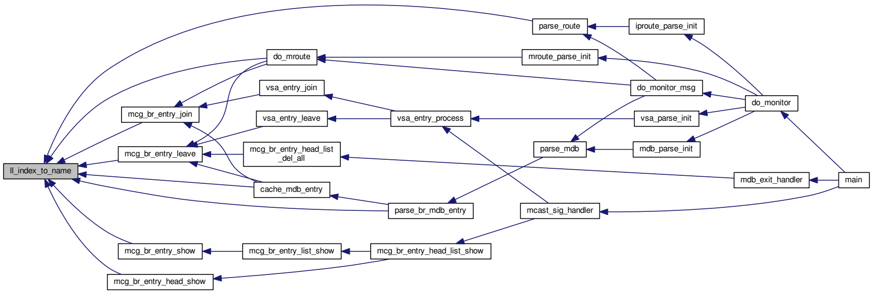
Here is the caller graph for this function:



6.2.3.14 const char∗ ll_index_to_name (unsigned *idx*)

Referenced by cache_mdb_entry(), do_mroute(), mcg_br_entry_head_show(), mcg_br_entry_join(), mcg_br_entry_leave(), mcg_br_entry_show(), parse_br_mdb_entry(), and parse_route().

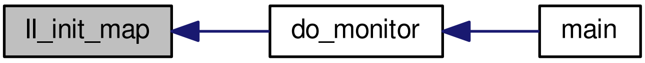
Here is the caller graph for this function:



6.2.3.15 void ll_init_map (struct rtnl_handle ∗ *rth*)

Referenced by do_monitor().

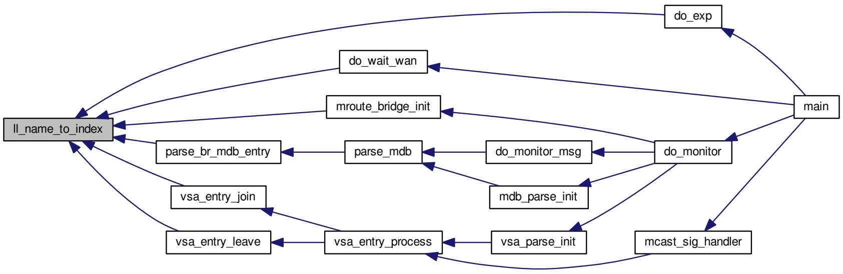
Here is the caller graph for this function:



6.2.3.16 unsigned ll_name_to_index (const char ∗ *name*)

Referenced by do_exp(), do_wait_wan(), mroute_bridge_init(), parse_br_mdb_entry(), vsa_entry_join(), and vsa_entry_leave().

Here is the caller graph for this function:



6.2.3.17 int main (int *argc,* char ∗∗ *argv*)

main entry point

parse argc argv and get stuff e.g. startup params etc.

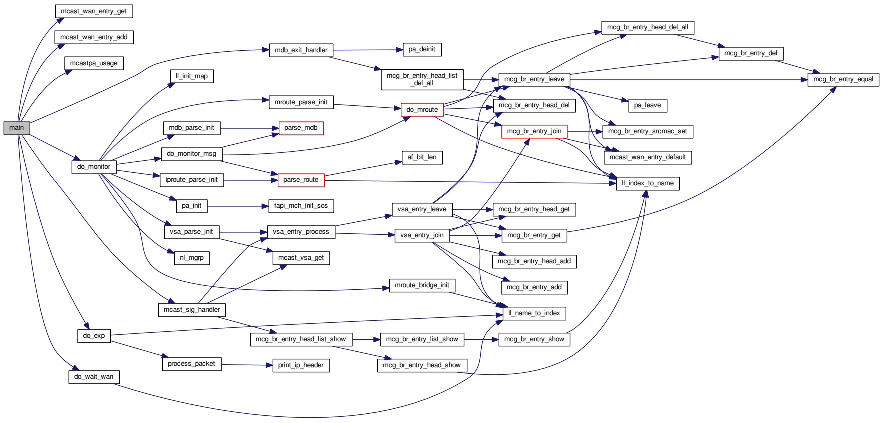
Note

Author

tim.hayes@smartrg.com

Definition at line 2374 of file mcast-pa.c.

Here is the call graph for this function:



6.2.3.18 struct mcast_ip_entry_t∗ mcast_ip_entry_add (char ∗ *address*)

add a ip address to our host list

Returns

pointer to entry or null

Note

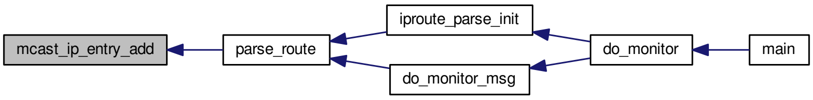
Author

tim.hayes@smartrg.com

Definition at line 540 of file mcast-pa.c.

Referenced by parse_route().

Here is the caller graph for this function:



6.2.3.19 int mcast_ip_entry_del (char ∗ *address*)

delete a ip address if found on list

Note

Author

tim.hayes@smartrg.com

Definition at line 563 of file mcast-pa.c.

6.2.3.20 struct mcast_ip_entry_t∗ mcast_ip_entry_get (char ∗ *address*)

returns a list entry of the ip address

removes instance of a group as a sub of the group head

Note

Returns

entry if address found null otherwise

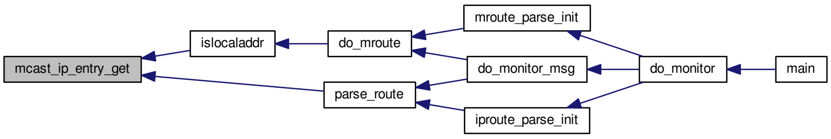
Author

tim.hayes@smartrg.com

Definition at line 590 of file mcast-pa.c.

Referenced by islocaladdr(), and parse_route().

Here is the caller graph for this function:



6.2.3.21 int mcast_ip_entry_list (void)

list all host ip addresses

Note

Author

tim.hayes@smartrg.com

Definition at line 632 of file mcast-pa.c.

6.2.3.22 void mcast_sig_handler (int *signo*)

sighandler

Note

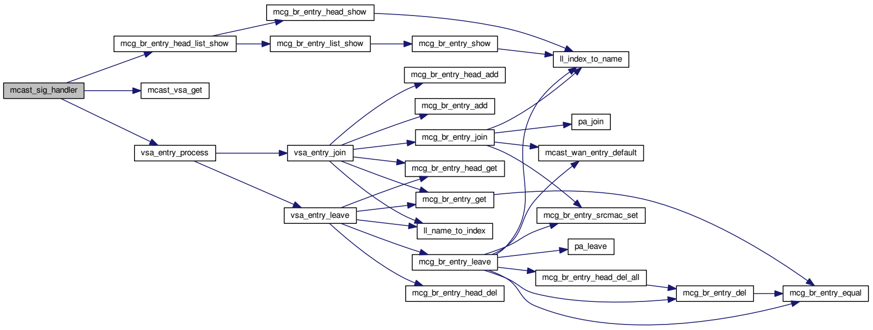
Author

tim.hayes@smartrg.com

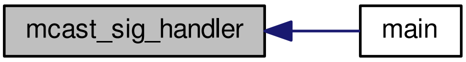
Definition at line 2189 of file mcast-pa.c.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.23 int mcast_vsa_get (void)

read a file and look for a vsa operation

join or leave channel device

Returns

0 if OK

Note

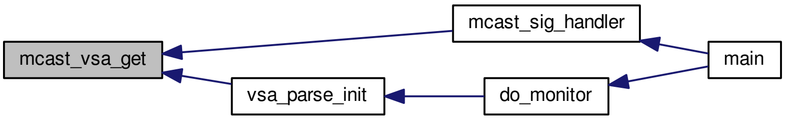
Author

tim.hayes@smartrg.com

Definition at line 351 of file mcast-pa.c.

Referenced by mcast_sig_handler(), and vsa_parse_init().

Here is the caller graph for this function:



6.2.3.24 struct mcast_wan_entry_t∗ mcast_wan_entry_add (char ∗ *name*)

add a wan iface to our host list

Returns

pointer to entry or null

Note

Author

tim.hayes@smartrg.com

Definition at line 376 of file mcast-pa.c.

Referenced by main().

Here is the caller graph for this function:

mcast_wan_entry_add
main

6.2.3.25 char∗ mcast_wan_entry_default (void)

returns a first entry of the wan iface

Note

Returns

first if name found null otherwise

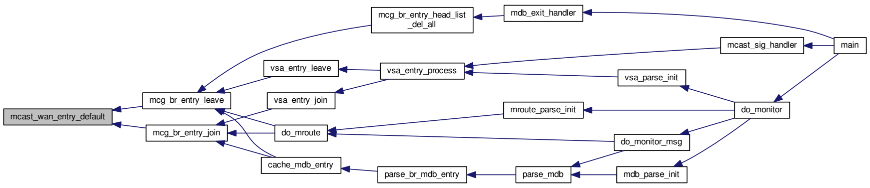
Author

tim.hayes@smartrg.com

Definition at line 451 of file mcast-pa.c.

Referenced by mcg_br_entry_join(), and mcg_br_entry_leave().

Here is the caller graph for this function:



6.2.3.26 int mcast_wan_entry_del (char ∗ *name*)

delete a wan iface if found on list

Note

Author

tim.hayes@smartrg.com

Definition at line 399 of file mcast-pa.c.

6.2.3.27 struct **mcast_wan_entry_t**∗ mcast_wan_entry_get (char ∗ *name*)

returns a list entry of the wan iface

removes instance of a wan iface

Note

Returns

entry if name found null otherwise

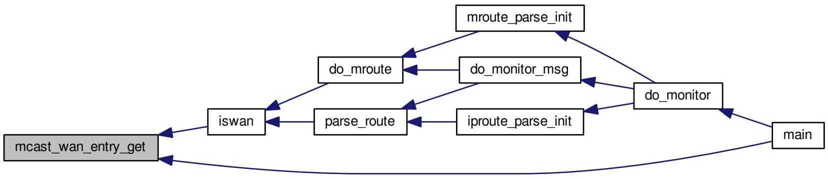
Author

tim.hayes@smartrg.com

Definition at line 426 of file mcast-pa.c.

Referenced by iswan(), and main().

Here is the caller graph for this function:



6.2.3.28 int mcast_wan_entry_list (void)

list all host wan ifaces

Note

Author

tim.hayes@smartrg.com

Definition at line 474 of file mcast-pa.c.

6.2.3.29 void mcastpa_usage (void)

usage helper

Note

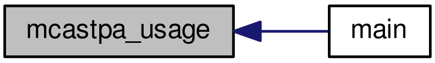
Author

tim.hayes@smartrg.com

Definition at line 2341 of file mcast-pa.c.

Referenced by main().

Here is the caller graph for this function:



6.2.3.30 struct mcg_br_mdb_entry_t∗ mcg_br_entry_add (struct mcg_br_mdb_entry_t ∗ *head,* struct br_mdb_entry ∗ *e*)

add a bridge mdb entry to the mc group list head list

subordinate members of group

Returns

pointer to entry or null

Note

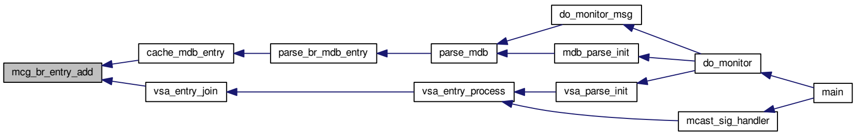
Author

tim.hayes@smartrg.com

Definition at line 774 of file mcast-pa.c.

Referenced by cache_mdb_entry(), and vsa_entry_join().

Here is the caller graph for this function:



6.2.3.31 int mcg_br_entry_del (struct mcg_br_mdb_entry_t ∗ *head,* struct br_mdb_entry ∗ *e*)

delete a bridge mdb entry from the mc group list head list

removes instance of a group as a sub of the group head

Note

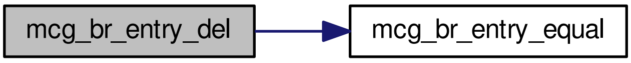
Author

tim.hayes@smartrg.com

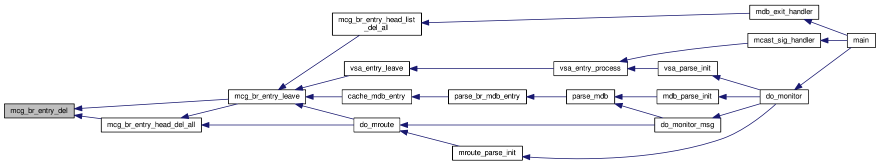
Definition at line 849 of file mcast-pa.c.

Referenced by mcg_br_entry_head_del_all(), and mcg_br_entry_leave().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.32 int mcg_br_entry_equal (struct br_mdb_entry ∗ *ex,* struct br_mdb_entry ∗ *ey*)

compare bridge entries

return 1 if equal 0 otherwise

Note

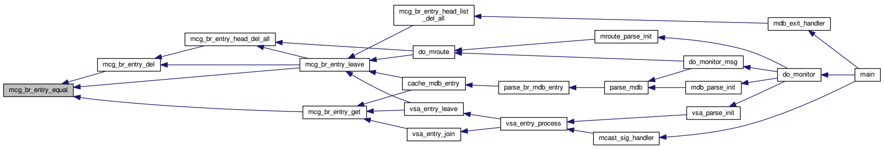
Author

tim.hayes@smartrg.com

Definition at line 698 of file mcast-pa.c.

Referenced by mcg_br_entry_del(), mcg_br_entry_get(), and mcg_br_entry_leave().

Here is the caller graph for this function:



6.2.3.33 struct mcg_br_mdb_entry_t∗ mcg_br_entry_get (struct mcg_br_mdb_entry_t ∗ *head,* struct br_mdb_entry ∗ *e*)

get a mdb entry from the mc group list head list of attached group entries

compares br_mdb_entries

Returns

pointer to entry or null

Note

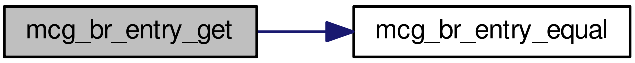
Author

tim.hayes@smartrg.com

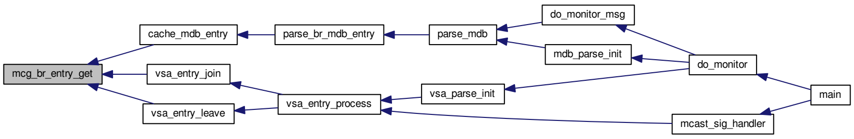
Definition at line 750 of file mcast-pa.c.

Referenced by cache_mdb_entry(), vsa_entry_join(), and vsa_entry_leave().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.34 struct mcg_br_mdb_entry_t∗ mcg_br_entry_head_add (struct br_mdb_entry ∗ *e*)

add a bridge mdb entry to the mc group list head

initial instance of a group

Returns

pointer to entry or null

Note

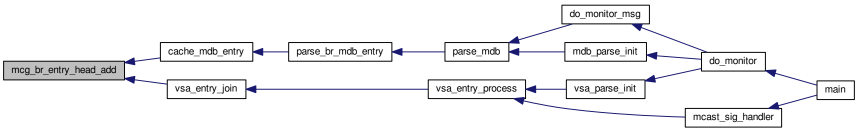
Author

tim.hayes@smartrg.com

Definition at line 798 of file mcast-pa.c.

Referenced by cache_mdb_entry(), and vsa_entry_join().

Here is the caller graph for this function:



6.2.3.35 int mcg_br_entry_head_del (struct **mcg_br_mdb_entry_t** ∗ *head*)

delete a bridge mdb entry from the mc group list head

removes instance of a group

Note

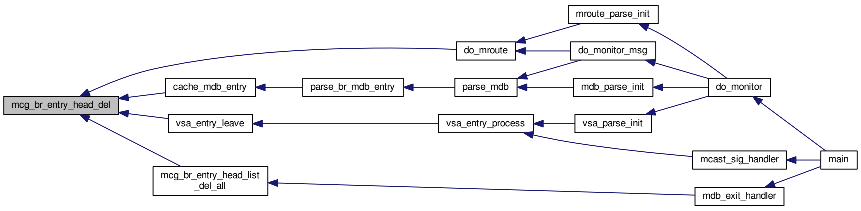
Author

tim.hayes@smartrg.com

Definition at line 823 of file mcast-pa.c.

Referenced by cache_mdb_entry(), do_mroute(), mcg_br_entry_head_list_del_all(), and vsa_entry_leave().

Here is the caller graph for this function:



6.2.3.36 void mcg_br_entry_head_del_all (struct **mcg_br_mdb_entry_t** ∗ *head*)

delete a bridge mdb entry from the mc group list head

removes instance of a group

Note

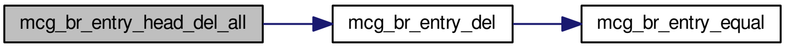
Author

tim.hayes@smartrg.com

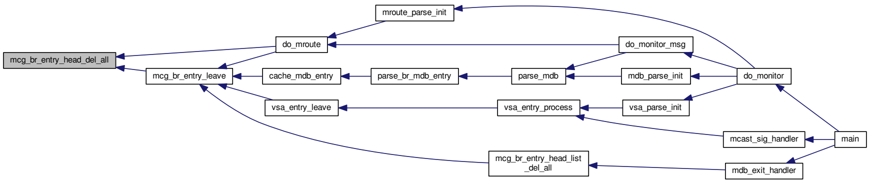
Definition at line 875 of file mcast-pa.c.

Referenced by do_mroute(), and mcg_br_entry_leave().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.37 struct mcg_br_mdb_entry_t∗ mcg_br_entry_head_get (struct br_mdb_entry ∗ *e*)

gets a head instance of mc group

just compares mcgroup

Returns

pointer to entry or null

Note

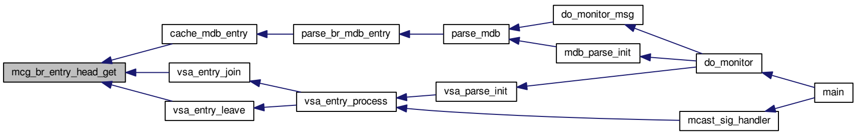
Author

tim.hayes@smartrg.com

Definition at line 720 of file mcast-pa.c.

Referenced by cache_mdb_entry(), vsa_entry_join(), and vsa_entry_leave().

Here is the caller graph for this function:



6.2.3.38 struct mcg_br_mdb_entry_t∗ mcg_br_entry_head_get_from_group (char ∗ *group*)

finds a head entry from a specific mc group

Returns

pointer if group found NULL otherwise

Note

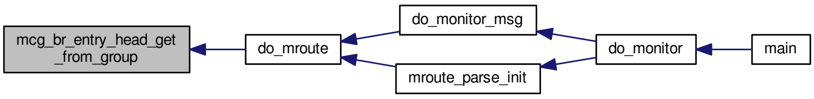
Author

tim.hayes@smartrg.com

Definition at line 1229 of file mcast-pa.c.

Referenced by do_mroute().

Here is the caller graph for this function:



6.2.3.39 int mcg_br_entry_head_list_del_all (void)

traverses all head groups, removes all sub entries issues leave and deletes head

calls leave for head group

Note

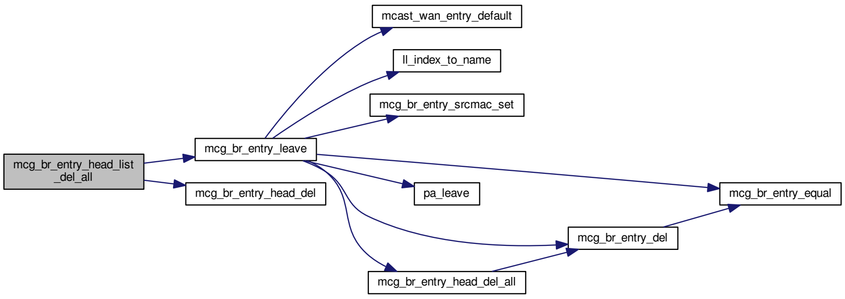
Author

tim.hayes@smartrg.com

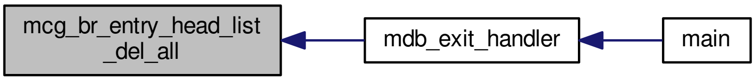
Definition at line 896 of file mcast-pa.c.

Referenced by mdb_exit_handler().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.40 void mcg_br_entry_head_list_show (void)

lists instances of head a mc group entires

Note

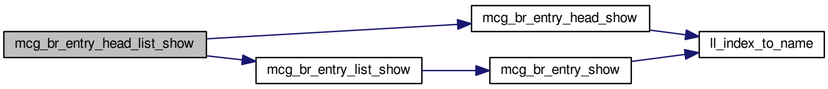
Author

tim.hayes@smartrg.com

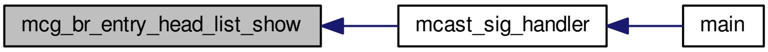
Definition at line 1107 of file mcast-pa.c.

Referenced by mcast_sig_handler().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.41 void mcg_br_entry_head_show (FILE ∗ *f*, struct mcg_br_mdb_entry_t ∗ *head*)

show an instance of a mc group entry head

Note

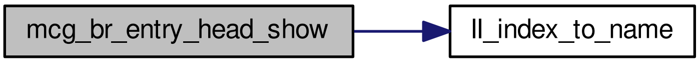
Author

tim.hayes@smartrg.com

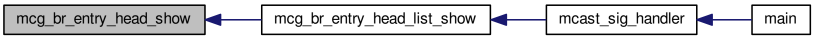
Definition at line 654 of file mcast-pa.c.

Referenced by mcg_br_entry_head_list_show().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.42 int mcg_br_entry_join (struct **mcg_br_mdb_entry_t** ∗ *head*)

joins a new group

calls hw specific pa_join()

Todo IPV6

Note

Author

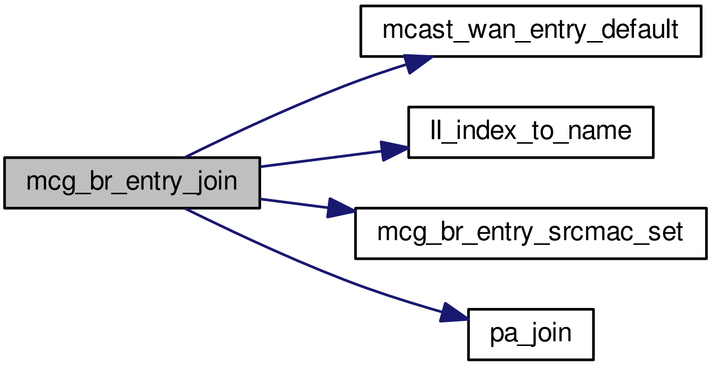
tim.hayes@smartrg.com

< ip address of video source - head use only

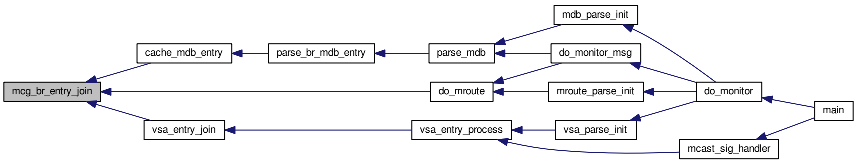
Definition at line 956 of file mcast-pa.c.

Referenced by cache_mdb_entry(), do_mroute(), and vsa_entry_join().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.43 int mcg_br_entry_leave (struct mcg_br_mdb_entry_t * *head,* struct br_mdb_entry * *e*)

leaves with no groups or joins with smaller group list

calls hw specific pa_leave()

Note

Author

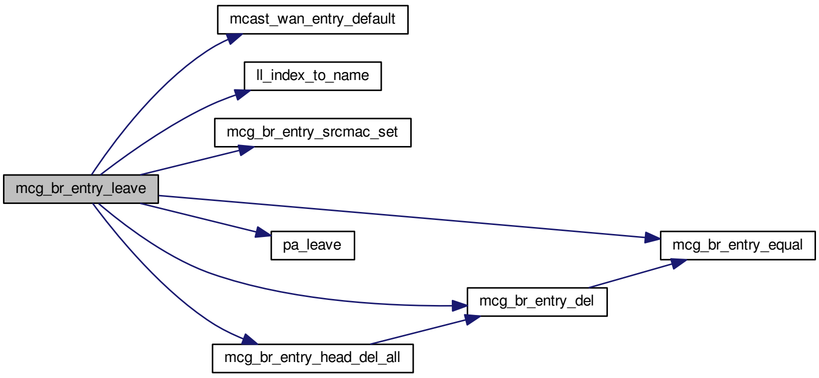
tim.hayes@smartrg.com

< ip address of video source - head use only

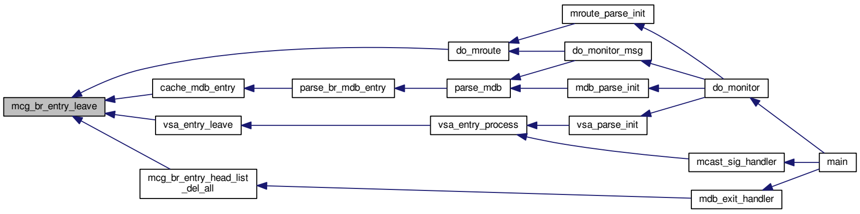
Definition at line 1029 of file mcast-pa.c.

Referenced by cache_mdb_entry(), do_mroute(), mcg_br_entry_head_list_del_all(), and vsa_entry_leave().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.44 void mcg_br_entry_list_show (FILE ∗ *f*, struct **mcg_br_mdb_entry_t** ∗ *head*)

lists instances of a mc group entires tied to a mc group head entry

Note

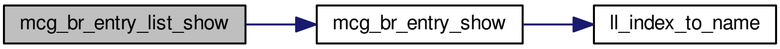
Author

tim.hayes@smartrg.com

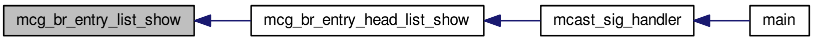
Definition at line 918 of file mcast-pa.c.

Referenced by mcg_br_entry_head_list_show().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.45 void mcg_br_entry_show (FILE ∗ *f,* struct mcg_br_mdb_entry_t ∗ *mcge*)

show an instance of a mc group entry

Note

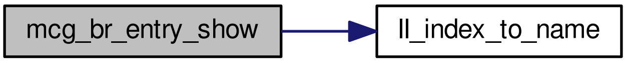
Author

tim.hayes@smartrg.com

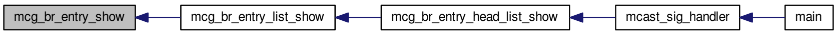
Definition at line 677 of file mcast-pa.c.

Referenced by mcg_br_entry_list_show().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.46 void mcg_br_entry_srcmac_set (struct mcg_br_mdb_entry_t ∗ *mcge,* struct mcastpa_join_leave_t ∗ *mjl*)

sets src mac in mjl struct

uses part of ipv6 union - this is supported by a bridge patch

Todo someday get a really clean way of doing this

Note

done to support Intel wifi mc which requires source of video subscriber

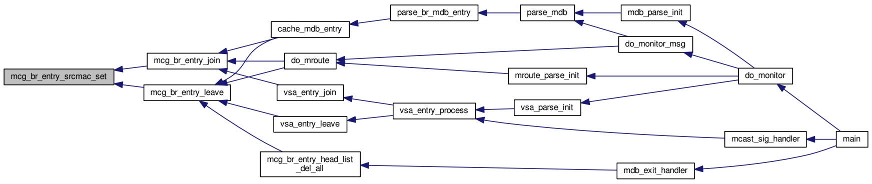
Author

tim.hayes@smartrg.com

Definition at line 941 of file mcast-pa.c.

Referenced by mcg_br_entry_join(), and mcg_br_entry_leave().

Here is the caller graph for this function:

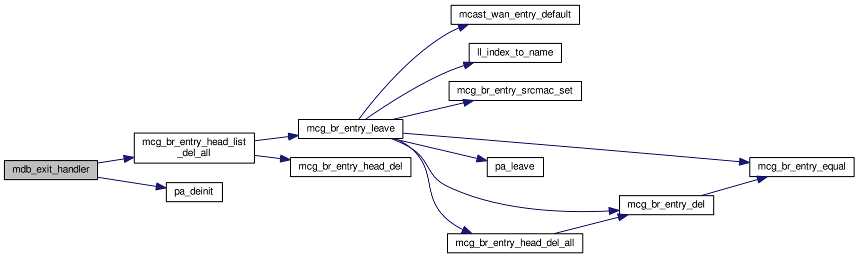


6.2.3.47 static void mdb_exit_handler (int *ev,* void ∗ *arg*) [static]

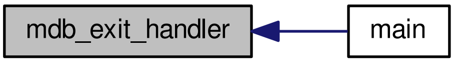
Definition at line 1254 of file mcast-pa.c.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.48 static int mdb_parse_init (void) [static]

initialize our lan mdb knowledge

Note

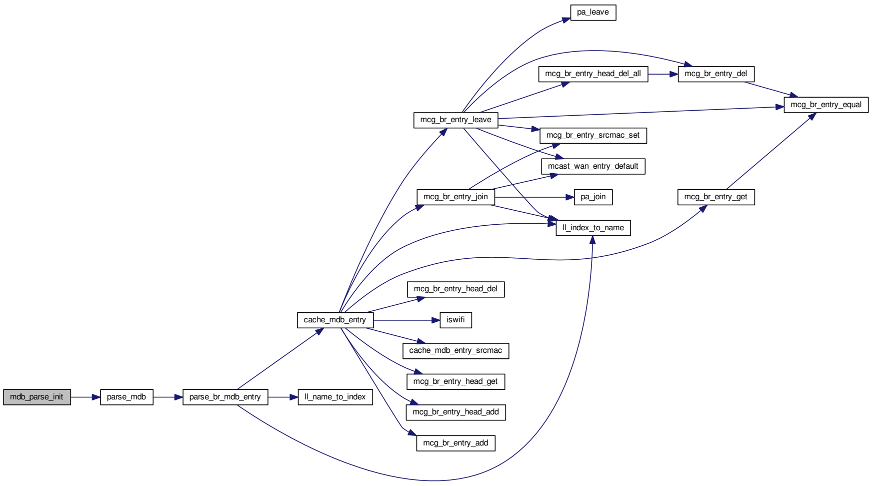
Author

tim.hayes@smartrg.com

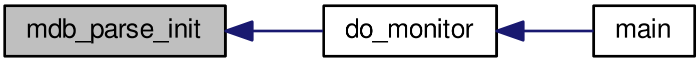
Definition at line 2057 of file mcast-pa.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:

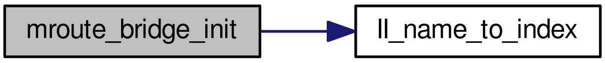


6.2.3.49 void mroute_bridge_init (void)

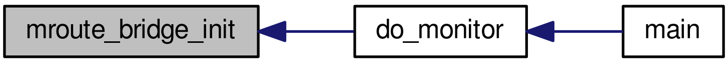
Definition at line 2010 of file mcast-pa.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.50 static int mroute_parse_init (void) [static]

initialize our wan mroute knowledge

Note

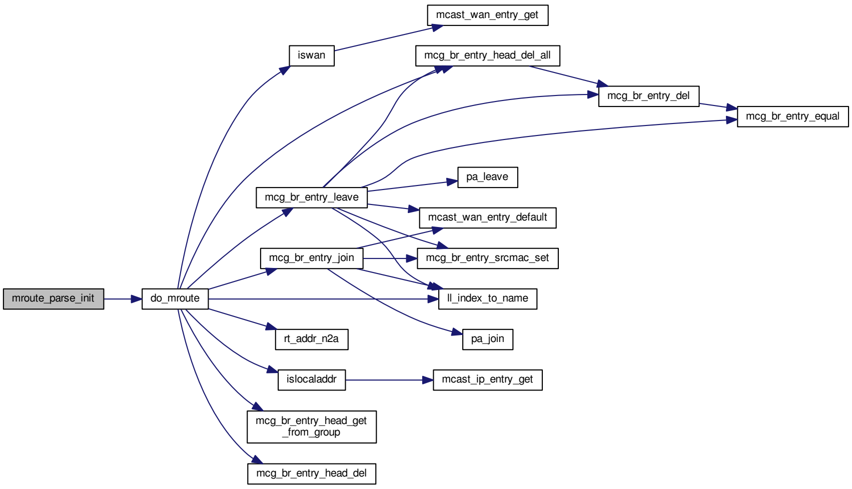
Author

tim.hayes@smartrg.com

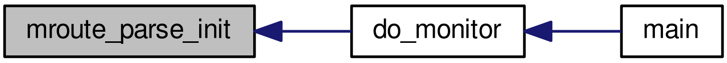
Definition at line 1985 of file mcast-pa.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:

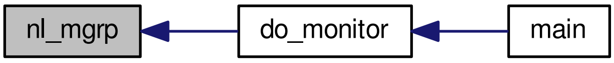


6.2.3.51 static __u32 nl_mgrp (__u32 *group*) [inline], [static]

Definition at line 332 of file mcast-pa.c.

Referenced by do_monitor().

Here is the caller graph for this function:



6.2.3.52 static void parse_br_mdb_entry (struct nlmsghdr ∗ *n,* int *ifindex,* struct rtattr ∗ *attr*) [static]

parses mdb_entry - may contain mulitiples and calls cache function

checks for bridge membership and not wan interface

Note

> We saw Cspire join on wan interface
> ifindex is bridge ifindex

Todo we only look for one wan interface.

what about guest bridge - need to check for that

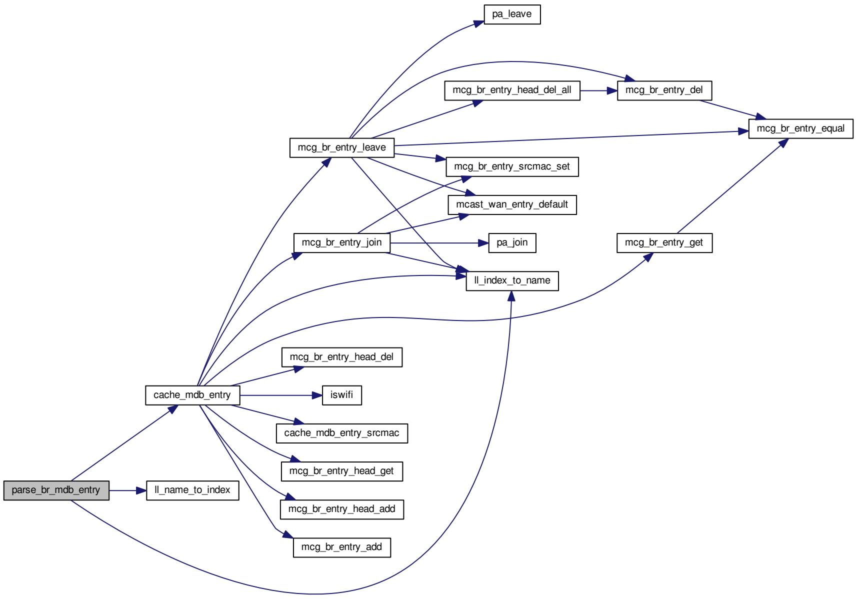
Author

> tim.hayes@smartrg.com

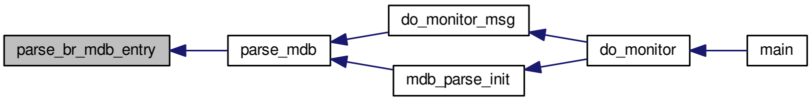
Definition at line 1770 of file mcast-pa.c.

Referenced by parse_mdb().

Here is the call graph for this function:



Here is the caller graph for this function:

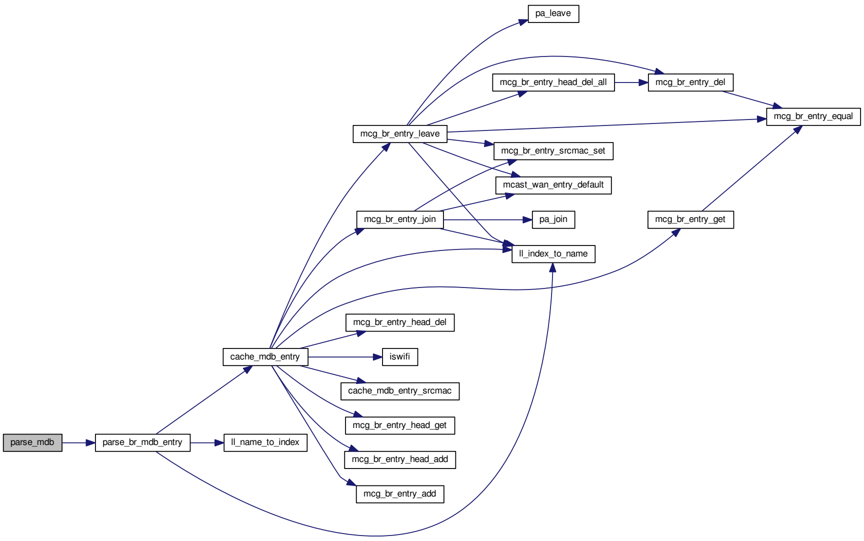


6.2.3.53 int parse_mdb (const struct sockaddr_nl ∗ *who,* struct nlmsghdr ∗ *n,* void ∗ *arg*)

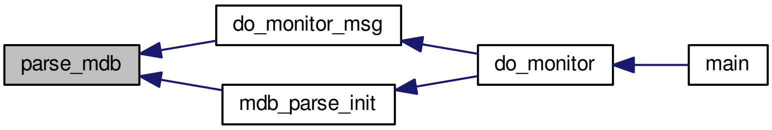
Definition at line 1799 of file mcast-pa.c.

Referenced by do_monitor_msg(), and mdb_parse_init().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.54 int parse_route (const struct sockaddr_nl ∗ *who,* struct nlmsghdr ∗ *n,* void ∗ *arg*)

parses route update and gets our host ip address

Todo support for multihoming and review

Note

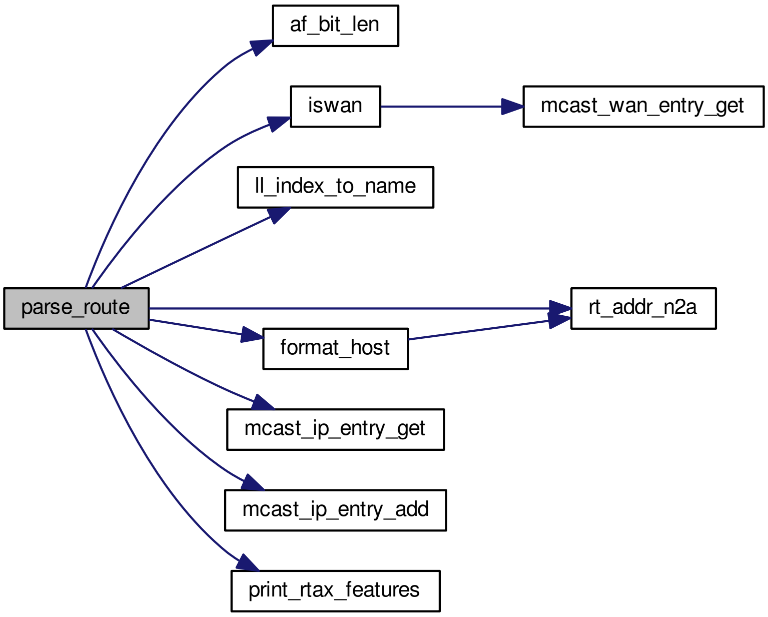
Author

tim.hayes@smartrg.com

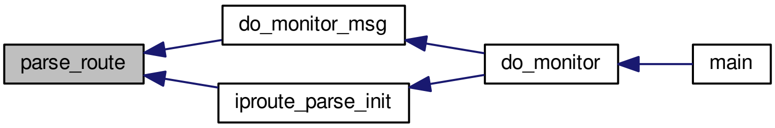
Definition at line 1375 of file mcast-pa.c.

Referenced by do_monitor_msg(), and iproute_parse_init().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.55 void print_ip_header (unsigned char ∗ *Buffer,* int *Size*)

experimental code here

Note

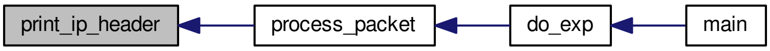
Author

tim.hayes@smartrg.com

Definition at line 2221 of file mcast-pa.c.

Referenced by process_packet().

Here is the caller graph for this function:



6.2.3.56 static void print_rtax_features (FILE ∗ *fp*, unsigned int *features*) [static]

misc parse_route support function

Note

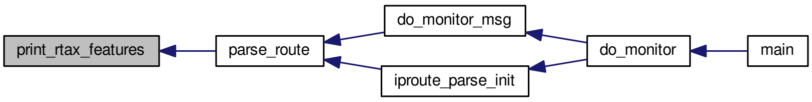
Author

tim.hayes@smartrg.com

Definition at line 1352 of file mcast-pa.c.

Referenced by parse_route().

Here is the caller graph for this function:

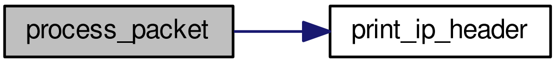


6.2.3.57 void process_packet (unsigned char ∗ *buffer*, int *size*)

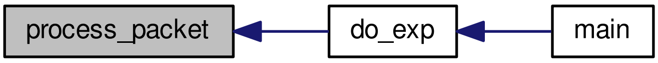
Definition at line 2247 of file mcast-pa.c.

Referenced by do_exp().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.58 const char∗ rt_addr_n2a (int *af,* const void ∗ *addr,* char ∗ *buf,* int *buflen*)

misc parse_route support function

Note

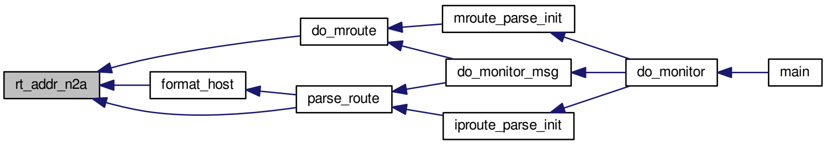
Author

tim.hayes@smartrg.com

Definition at line 1276 of file mcast-pa.c.

Referenced by do_mroute(), format_host(), and parse_route().

Here is the caller graph for this function:



6.2.3.59 static int rtm_get_table (struct rtmsg ∗ *r,* struct rtattr ∗∗ *tb*) [inline], [static]

misc parse_route support function

Note

Author

tim.hayes@smartrg.com

Definition at line 1335 of file mcast-pa.c.

6.2.3.60 void vsa_entry_join (void)

joins a vsa requested group on a device

emulates a bridge device joining

Note

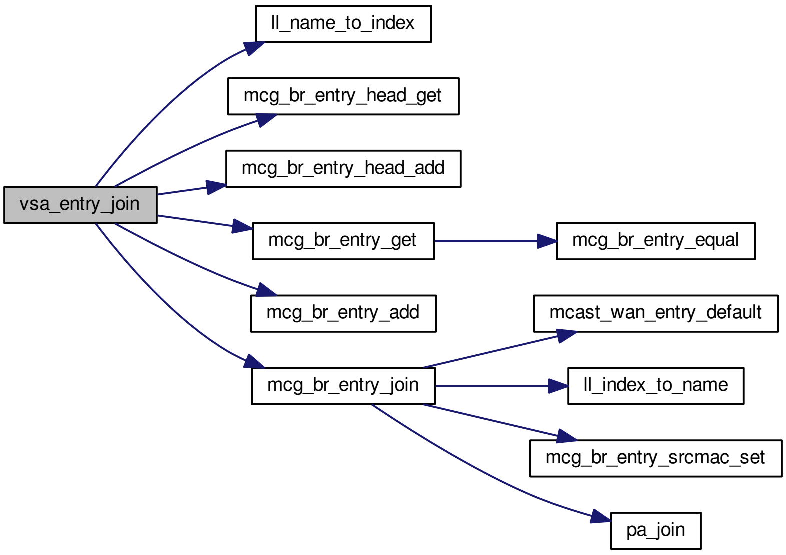
Author

tim.hayes@smartrg.com

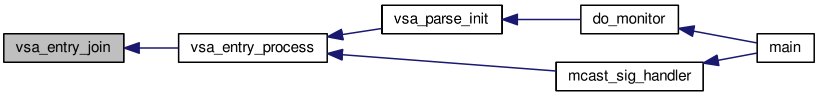
Definition at line 1134 of file mcast-pa.c.

Referenced by vsa_entry_process().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.61 void vsa_entry_leave (void)

leave a vsa requested group on a device

emulates a bridge device joining

Note

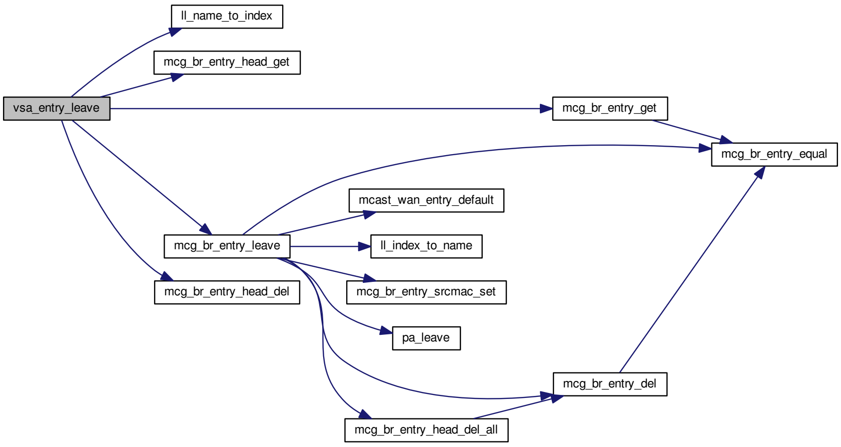
Author

tim.hayes@smartrg.com

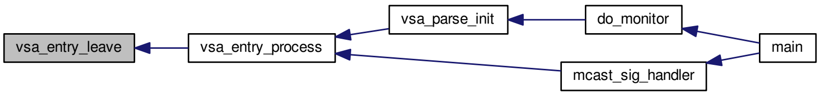
Definition at line 1170 of file mcast-pa.c.

Referenced by vsa_entry_process().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.62 void vsa_entry_process (void)

process a vsa request

Note

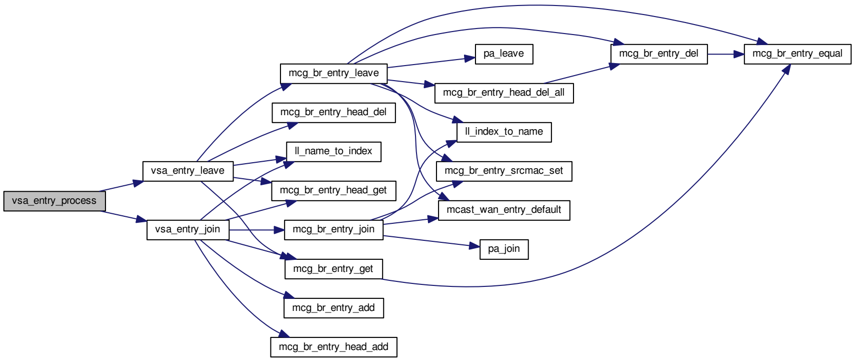
Author

tim.hayes@smartrg.com

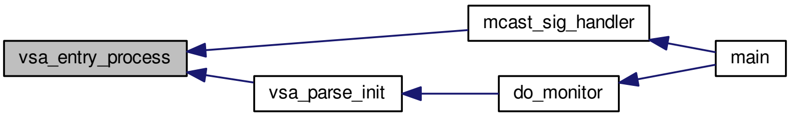
Definition at line 1209 of file mcast-pa.c.

Referenced by mcast_sig_handler(), and vsa_parse_init().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.63 static int vsa_parse_init (void) [static]

handle any vsa requests

Note

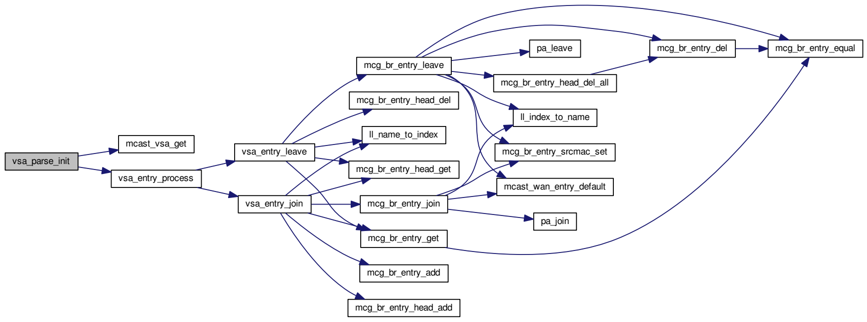
Author

tim.hayes@smartrg.com

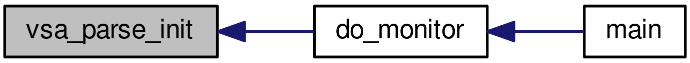
Definition at line 2081 of file mcast-pa.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.4 Variable Documentation

6.2.4.1 char∗ _SL_ = "\n"

Definition at line 227 of file mcast-pa.c.

Referenced by parse_route().

6.2.4.2 struct option long_options[] [static]

Initial value:

```
= {
    {"verbose", no_argument, 0, 'v'},
    {"foreground", no_argument, 0, 'f'},
    {"wan", required_argument, 0, 'w'},
    {"bridge", required_argument, 0, 'b'},
    {"video2lan", required_argument, 0, 'V'},
    {"src", required_argument, 0, 's'},
    {"exp", no_argument, 0, 'x'},
    {"nowifi", no_argument, 0, 'n'},
    {0, 0, 0, 0}
}
```

Definition at line 2353 of file mcast-pa.c.

Referenced by main().

6.2.4.3 struct mcastpa_t mcastpa [static]

Definition at line 260 of file mcast-pa.c.

Referenced by cache_mdb_entry(), do_monitor(), do_wait_wan(), iswan(), main(), mcast_ip_entry_add(), mcast_ip_entry_del(), mcast_ip_entry_get(), mcast_ip_entry_list(), mcast_sig_handler(), mcast_vsa_get(), mcast_wan_entry_add(), mcast_wan_entry_default(), mcast_wan_entry_del(), mcast_wan_entry_get(), mcast_wan_entry_list(), mcg_br_entry_head_add(), mcg_br_entry_head_del(), mcg_br_entry_head_del_all(), mcg_br_entry_head_get(), mcg_br_entry_head_get_from_group(), mcg_br_entry_head_list_del_all(), mcg_br_entry_head_list_show(), mcg_br_entry_join(), mcg_br_entry_leave(), parse_br_mdb_entry(), vsa_entry_join(), vsa_entry_leave(), vsa_entry_process(), and vsa_parse_init().

6.2.4.4 const char∗ mx_names[RTAX_MAX+1] [static]

Initial value:

```
= {
    [RTAX_MTU] = "mtu",
    [RTAX_WINDOW] = "window",
    [RTAX_RTT] = "rtt",
    [RTAX_RTTVAR] = "rttvar",
    [RTAX_SSTHRESH] = "ssthresh",
    [RTAX_CWND] = "cwnd",
    [RTAX_ADVMSS] = "advmss",
    [RTAX_REORDERING] = "reordering",
    [RTAX_HOPLIMIT] = "hoplimit",
    [RTAX_INITCWND] = "initcwnd",
    [RTAX_FEATURES] = "features",
    [RTAX_RTO_MIN] = "rto_min",
    [RTAX_INITRWND] = "initrwnd",
}
```

Definition at line 236 of file mcast-pa.c.

Referenced by parse_route().

6.2.4.5 struct rtnl_handle rth = {.fd = -1 }

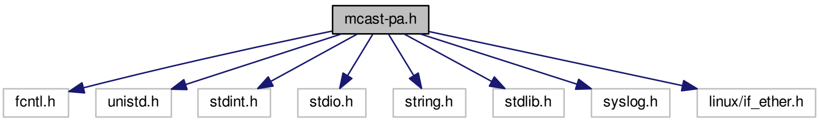
Definition at line 262 of file mcast-pa.c.

Referenced by do_monitor(), iproute_parse_init(), mdb_parse_init(), and mroute_parse_init().

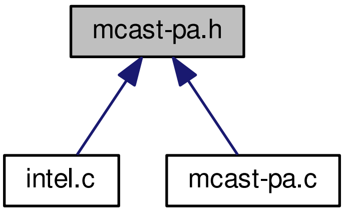
6.3 mcast-pa.h File Reference

```
#include <fcntl.h>
#include <unistd.h>
#include <stdint.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <syslog.h>
#include <linux/if_ether.h>
```

Include dependency graph for mcast-pa.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct mcastpa_system_init_t
- struct mcastpa_join_leave_t

Macros

- #define MCASTPA_STRING_SIZE 128
- #define MSI_FLAG_EXP 1<<0
- #define MJL_FLAG_EXP 1<<0
- #define MJL_FLAG_BRIDGE 1<<1
- #define MJL_FLAG_SRCIP 1<<2
- #define MJL_FLAG_LAN 1<<3
- #define MJL_FLAG_UPDATE 1<<4

Functions

- int pa_init (struct mcastpa_system_init_t ∗msi)

  inits intel mcast fapi subsystem using fapi_mcast.c (libmcastfapi)
- int pa_join (struct mcastpa_join_leave_t ∗mjl)

  joins by libmcastfapi call
- int pa_leave (struct mcastpa_join_leave_t ∗mjl)

  leaves by libmcastfapi call
- int pa_deinit (struct mcastpa_system_init_t ∗msi)

  de-inits intel mcast subsystem

6.3.1 Macro Definition Documentation

6.3.1.1 #define MCASTPA_STRING_SIZE 128

Definition at line 31 of file mcast-pa.h.

6.3.1.2 #define MJL_FLAG_BRIDGE 1<<1

bridge mode - no src ip

Definition at line 42 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), mcg_br_entry_leave(), and pa_join().

6.3.1.3 #define MJL_FLAG_EXP 1<<0

experimental use

Definition at line 41 of file mcast-pa.h.

6.3.1.4 #define MJL_FLAG_LAN 1<<3

contains lan entries i.e. not empty

Definition at line 44 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), and mcg_br_entry_leave().

6.3.1.5 #define MJL_FLAG_SRCIP 1<<2

srcip included

Definition at line 43 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), and mcg_br_entry_leave().

6.3.1.6 #define MJL_FLAG_UPDATE 1<<4

group has been joined at least once i.e. update to add

Definition at line 45 of file mcast-pa.h.

Referenced by mcg_br_entry_join(), and pa_join().

6.3.1.7 #define MSI_FLAG_EXP 1<<0

Definition at line 34 of file mcast-pa.h.

6.3.2 Function Documentation

6.3.2.1 int pa_deinit (struct mcastpa_system_init_t ∗ *msi*)

de-inits intel mcast subsystem

not used because it unload the module

Note

Todo flush entries

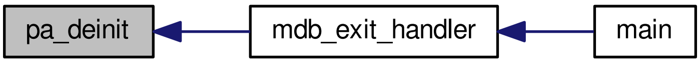
Author

tim.hayes@smartrg.com

Definition at line 416 of file intel.c.

Referenced by mdb_exit_handler().

Here is the caller graph for this function:



6.3.2.2 int pa_init (struct mcastpa_system_init_t * *msi*)

inits intel mcast fapi subsystem using fapi_mcast.c (libmcastfapi)

Note

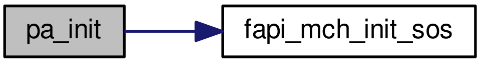
Author

tim.hayes@smartrg.com

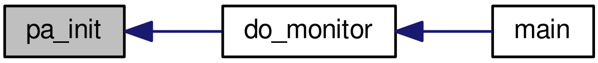
Definition at line 317 of file intel.c.

Referenced by do_monitor().

Here is the call graph for this function:



Here is the caller graph for this function:



6.3.2.3 int pa_join (struct mcastpa_join_leave_t ∗ *mjl*)

joins by libmcastfapi call

first group join is add additions are updates

Note

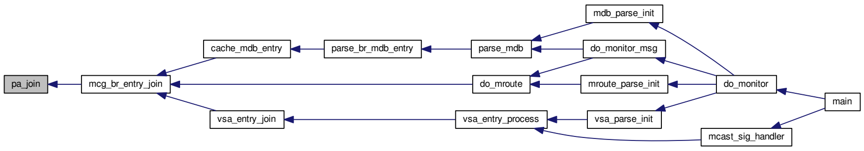
Author

tim.hayes@smartrg.com

Definition at line 340 of file intel.c.

Referenced by mcg_br_entry_join().

Here is the caller graph for this function:



6.3.2.4 int pa_leave (struct mcastpa_join_leave_t ∗ *mjl*)

leaves by libmcastfapi call

all group leaves are deletes no matter how many members

Note

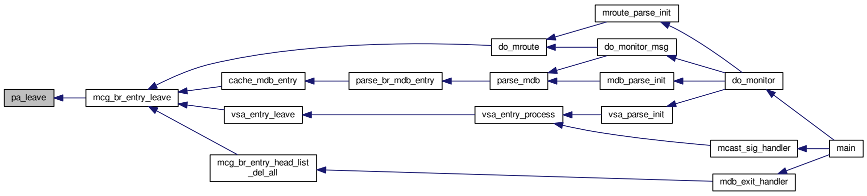
Author

tim.hayes@smartrg.com

Definition at line 387 of file intel.c.

Referenced by mcg_br_entry_leave().

Here is the caller graph for this function:



Index