

MCAST-PA

1.0.0

Generated by Doxygen 1.8.6

Tue Jul 17 2018 09:46:41

Contents

1	Multicast Packet Accelerator	1
1.1	Overview	1
1.1.1	Invocation	2
1.1.2	Logging	2
1.1.3	MCPROXY	2
2	Todo List	5
3	Data Structure Index	7
3.1	Data Structures	7
4	File Index	9
4.1	File List	9
5	Data Structure Documentation	11
5.1	mcast_ip_entry_t Struct Reference	11
5.1.1	Detailed Description	11
5.1.2	Field Documentation	11
5.1.2.1	address	11
5.1.2.2	head	11
5.1.2.3	ifindex	11
5.1.2.4	name	11
5.2	mcast_wan_entry_t Struct Reference	12
5.2.1	Detailed Description	12
5.2.2	Field Documentation	12
5.2.2.1	address	12
5.2.2.2	head	12
5.2.2.3	ifindex	12
5.2.2.4	name	12
5.3	mcastpa_join_leave_t Struct Reference	13
5.3.1	Detailed Description	13
5.3.2	Field Documentation	13
5.3.2.1	flags	13

5.3.2.2	group	13
5.3.2.3	lan	13
5.3.2.4	lan_dev	13
5.3.2.5	srcip	14
5.3.2.6	srcmac	14
5.3.2.7	wan	14
5.4	mcastpa_system_init_t Struct Reference	14
5.4.1	Detailed Description	14
5.4.2	Field Documentation	14
5.4.2.1	flags	14
5.4.2.2	srcip	14
5.4.2.3	wan	15
5.5	mcastpa_t Struct Reference	15
5.5.1	Detailed Description	15
5.5.2	Field Documentation	16
5.5.2.1	current_time	16
5.5.2.2	dev_name	16
5.5.2.3	error_ctime	16
5.5.2.4	group	16
5.5.2.5	ip_head	16
5.5.2.6	last_time	16
5.5.2.7	mcg_head	16
5.5.2.8	params	16
5.5.2.9	sock	17
5.5.2.10	start_ctime	17
5.5.2.11	start_time	17
5.5.2.12	timer_tick	17
5.5.2.13	wan_head	17
5.6	mcg_br_mdb_entry_t Struct Reference	17
5.6.1	Detailed Description	17
5.6.2	Field Documentation	18
5.6.2.1	br_ifindex	18
5.6.2.2	e	18
5.6.2.3	joined	18
5.6.2.4	mcg_entry	18
5.6.2.5	mcg_head	18
5.6.2.6	src	18
5.6.2.7	wan_ifindex	18
5.7	params_t Struct Reference	19
5.7.1	Detailed Description	19

5.7.2	Field Documentation	20
5.7.2.1	bridge_name	20
5.7.2.2	bridged	20
5.7.2.3	dbg	20
5.7.2.4	exp	20
5.7.2.5	foreground	20
5.7.2.6	monitor	20
5.7.2.7	nowifi	20
5.7.2.8	src	21
5.7.2.9	use_src	21
5.7.2.10	verbose	21
5.7.2.11	video2lan	21
5.7.2.12	video2lan_name	21
5.7.2.13	vsa	21
5.7.2.14	wan	21
5.7.2.15	wan_ifindex	21
5.8	vsa_t Struct Reference	22
5.8.1	Detailed Description	22
5.8.2	Field Documentation	22
5.8.2.1	device	22
5.8.2.2	group	22
5.8.2.3	op	22
5.8.2.4	valid	22
6	File Documentation	23
6.1	intel.c File Reference	23
6.1.1	Detailed Description	24
6.1.2	Macro Definition Documentation	24
6.1.2.1	INTEL_MCAST_USE_MCAST_FAPI	24
6.1.2.2	MCAST_HELPER_DEV_MAJOR_NUM	24
6.1.2.3	MCAST_HELPER_DEV_MINOR_NUM	24
6.1.2.4	MCAST_HELPER_DEVICE	24
6.1.3	Function Documentation	24
6.1.3.1	fapi_mch_init_sos	24
6.1.3.2	pa_deinit	25
6.1.3.3	pa_init	26
6.1.3.4	pa_init_sos	26
6.1.3.5	pa_join	27
6.1.3.6	pa_leave	27
6.2	mcast-pa.c File Reference	28

6.2.1	Detailed Description	32
6.2.2	Macro Definition Documentation	32
6.2.2.1	CMD_BUF_SIZE	32
6.2.2.2	INET_ADDR_SIZE	32
6.2.2.3	MDB_FORK_EXIT	32
6.2.2.4	MDBA_RTA	32
6.2.2.5	MROUTE_DEFAULT_TTL	32
6.2.2.6	MROUTE_RATE_LIMIT_ENDLESS	33
6.2.2.7	MROUTE_TTL_THRESHOLD	33
6.2.2.8	PRTFL	33
6.2.2.9	SPRINT_BSIZE	33
6.2.2.10	SPRINT_BUF	33
6.2.3	Function Documentation	33
6.2.3.1	af_bit_len	33
6.2.3.2	cache_mdb_entry	34
6.2.3.3	cache_mdb_entry_srcmac	35
6.2.3.4	do_exp	35
6.2.3.5	do_monitor	35
6.2.3.6	do_monitor_msg	36
6.2.3.7	do_mroute	37
6.2.3.8	do_wait_wan	38
6.2.3.9	format_host	39
6.2.3.10	iproute_parse_init	40
6.2.3.11	islocaladdr	41
6.2.3.12	iswan	42
6.2.3.13	iswifi	42
6.2.3.14	ll_index_to_name	43
6.2.3.15	ll_init_map	43
6.2.3.16	ll_name_to_index	44
6.2.3.17	main	44
6.2.3.18	mcast_ip_entry_add	44
6.2.3.19	mcast_ip_entry_del	45
6.2.3.20	mcast_ip_entry_get	45
6.2.3.21	mcast_ip_entry_list	46
6.2.3.22	mcast_sig_handler	46
6.2.3.23	mcast_vsa_get	47
6.2.3.24	mcast_wan_entry_add	48
6.2.3.25	mcast_wan_entry_default	48
6.2.3.26	mcast_wan_entry_del	49
6.2.3.27	mcast_wan_entry_get	49

6.2.3.28	mcast_wan_entry_list	50
6.2.3.29	mcastpa_usage	50
6.2.3.30	mcg_br_entry_add	51
6.2.3.31	mcg_br_entry_del	51
6.2.3.32	mcg_br_entry_equal	52
6.2.3.33	mcg_br_entry_get	52
6.2.3.34	mcg_br_entry_head_add	53
6.2.3.35	mcg_br_entry_head_del	54
6.2.3.36	mcg_br_entry_head_del_all	54
6.2.3.37	mcg_br_entry_head_get	55
6.2.3.38	mcg_br_entry_head_get_from_group	55
6.2.3.39	mcg_br_entry_head_list_del_all	56
6.2.3.40	mcg_br_entry_head_list_show	57
6.2.3.41	mcg_br_entry_head_show	57
6.2.3.42	mcg_br_entry_join	58
6.2.3.43	mcg_br_entry_leave	59
6.2.3.44	mcg_br_entry_list_show	60
6.2.3.45	mcg_br_entry_show	61
6.2.3.46	mcg_br_entry_srcmac_set	62
6.2.3.47	mdb_exit_handler	62
6.2.3.48	mdb_parse_init	63
6.2.3.49	mroute_bridge_init	64
6.2.3.50	mroute_parse_init	64
6.2.3.51	nl_mgrp	65
6.2.3.52	parse_br_mdb_entry	66
6.2.3.53	parse_mdb	67
6.2.3.54	parse_route	67
6.2.3.55	print_ip_header	68
6.2.3.56	print_rtax_features	69
6.2.3.57	process_packet	69
6.2.3.58	rt_addr_n2a	70
6.2.3.59	rtm_get_table	70
6.2.3.60	vsa_entry_join	71
6.2.3.61	vsa_entry_leave	72
6.2.3.62	vsa_entry_process	73
6.2.3.63	vsa_parse_init	73
6.2.4	Variable Documentation	74
6.2.4.1	_SL_	74
6.2.4.2	long_options	74
6.2.4.3	mcastpa	75

6.2.4.4	<code>mx_names</code>	75
6.2.4.5	<code>rth</code>	75
6.3	<code>mcast-pa.h</code> File Reference	75
6.3.1	Macro Definition Documentation	76
6.3.1.1	<code>MCASTPA_STRING_SIZE</code>	76
6.3.1.2	<code>MJL_FLAG_BRIDGE</code>	77
6.3.1.3	<code>MJL_FLAG_EXP</code>	77
6.3.1.4	<code>MJL_FLAG_LAN</code>	77
6.3.1.5	<code>MJL_FLAG_SRCIP</code>	77
6.3.1.6	<code>MJL_FLAG_UPDATE</code>	77
6.3.1.7	<code>MSI_FLAG_EXP</code>	77
6.3.2	Function Documentation	77
6.3.2.1	<code>pa_deinit</code>	77
6.3.2.2	<code>pa_init</code>	78
6.3.2.3	<code>pa_join</code>	79
6.3.2.4	<code>pa_leave</code>	79
Index		80

Chapter 1

Multicast Packet Accelerator

1.1 Overview

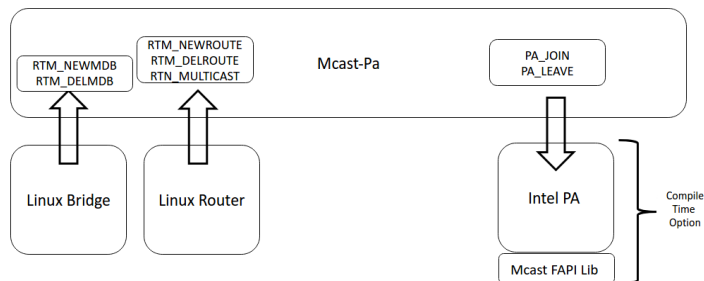
The basic OVRT multicast model consisting of the multicast aware Linux bridge and MCPROXY works on platforms where the CPU has the horsepower to meet the multicast delivery requirements and perform general CPE management functions in parallel.

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.

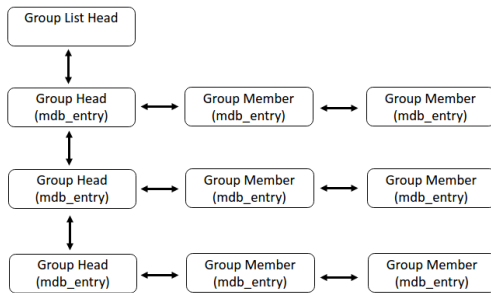
The RTNETLINK callbacks consist of two components - bridge MDB callbacks and router ROUTE callbacks. The bridge MDB callbacks consist of NEWMDB and DELMDB messages that result in join and leave operations respectively. The ROUTE callbacks consist of NEWROUTE and DELROUTE messages of type RTN_MULTICAST.

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 `mcbg_br_entry_join` (struct `mcbg_br_mdb_entry_t` *head)

IPV6

Note

Author

`tim.hayes@smartrg.com`

Global `mcbg_br_entry_srcmac_set` (struct `mcbg_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 `nlmsg_hdr` *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 `nlmsg_hdr` *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:

mcast_ip_entry_t	??
mcast_wan_entry_t	??
mcastpa_join_leave_t	??
mcastpa_system_init_t	??
mcastpa_t	??
mcg_br_mdb_entry_t	??
params_t	??
vsa_t	??

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

intel.c	Intel Multicast Packet Accelerator	??
mcast-pa.c	Multicast Packet Accelerator	??
mcast-pa.h	??

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

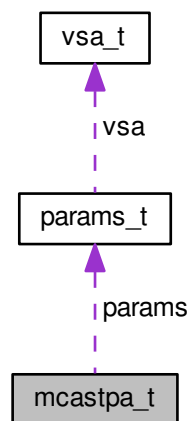
Definition at line 37 of file mcast-pa.h.

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.

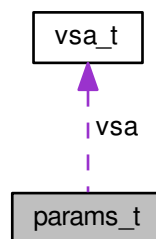
Referenced by `do_mroute()`, and `mcg_br_entry_head_show()`.

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

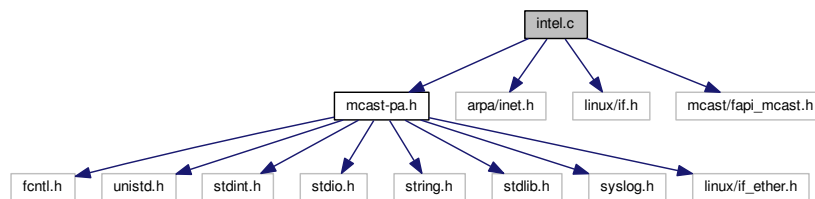
File Documentation

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

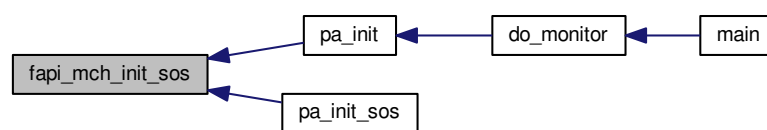
Author

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

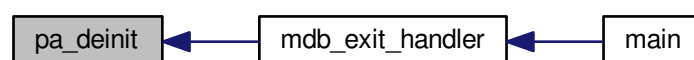
Author

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

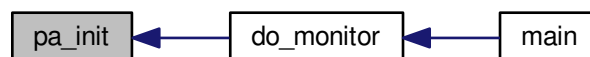
Definition at line 317 of file intel.c.

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

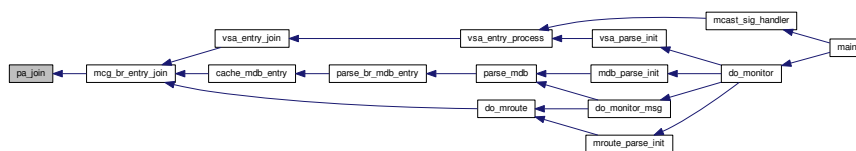
Author

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

- 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 nlmsgghdr *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 nlmsgghdr *n, int ifindex, struct br_mdb_entry *e)
- static void [parse_br_mdb_entry](#) (struct nlmsgghdr *n, int ifindex, struct rtattr *attr)
parses mdb_entry - may contain multiples and calls cache function
- int [parse_mdb](#) (const struct sockaddr_nl *who, struct nlmsgghdr *n, void *arg)
- int [do_mroute](#) (const struct sockaddr_nl *who, struct nlmsgghdr *n, void *arg)
- static int [do_monitor_msg](#) (const struct sockaddr_nl *who, struct nlmsgghdr *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

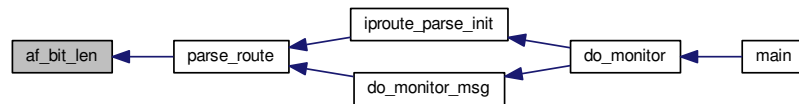
Author

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:

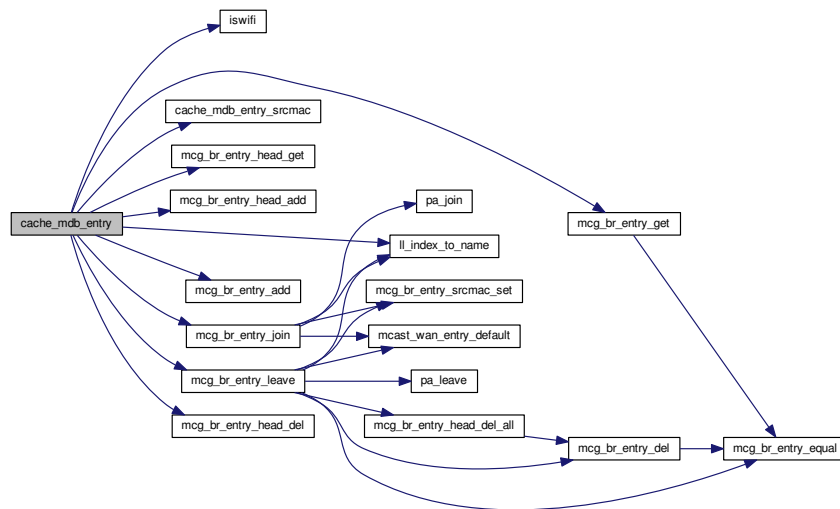


6.2.3.2 `static void cache_mdb_entry (struct nlmsg_hdr * n, int ifindex, struct br_mdb_entry * e)` [static]

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:

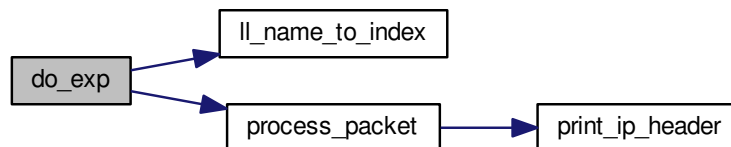


6.2.3.4 void do_exp (void)

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

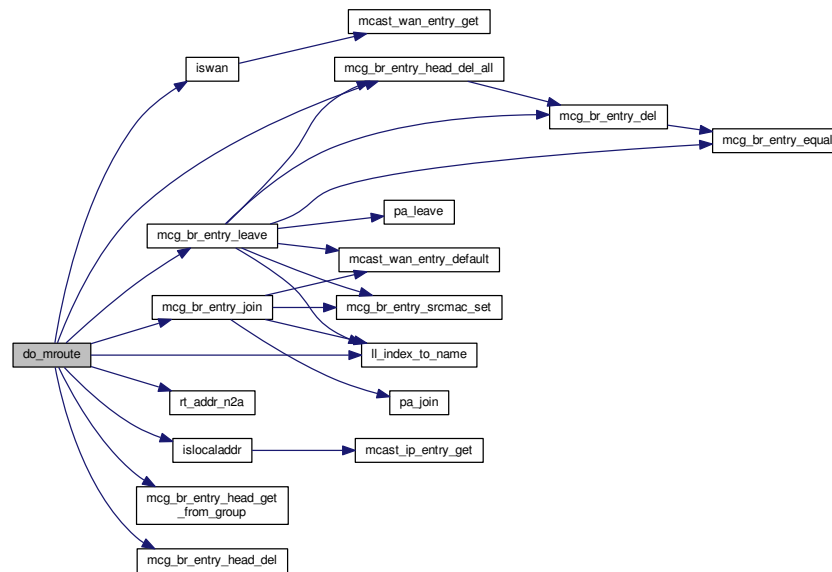
tim.hayes@smartrg.com

Referenced by main().

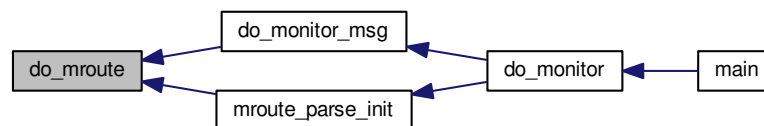
```
graph LR; main --> do_monitor
```

Referenced by `do_monitor()`.

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

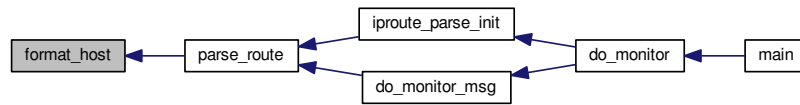
Definition at line 1296 of file mcast-pa.c.

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

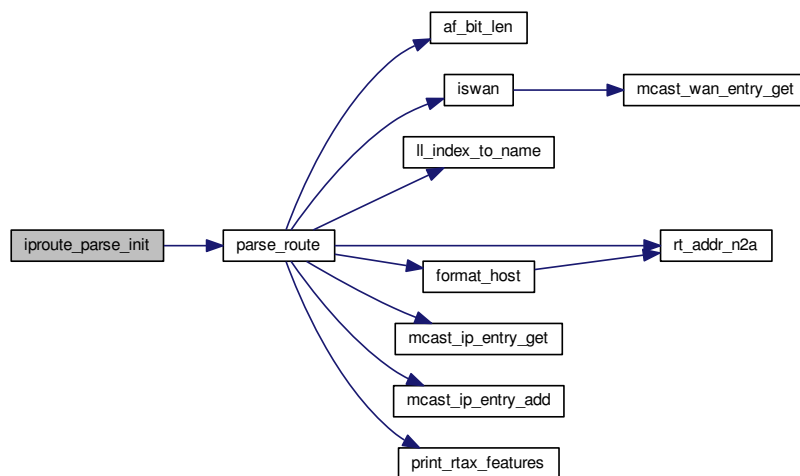
Author

tim.hayes@smartrg.com

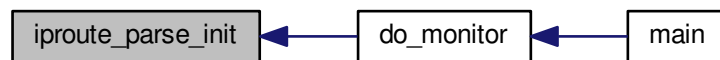
Definition at line 1652 of file mcast-pa.c.

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 addresses

Returns

1 if ours 0 otherwise

Note

Author

tim.hayes@smartrg.com

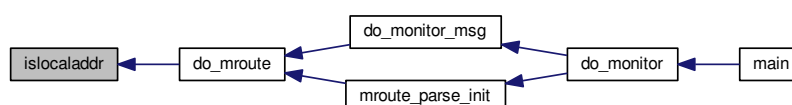
Definition at line 615 of file `mcast-pa.c`.

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

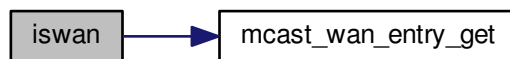
Author

tim.hayes@smartrg.com

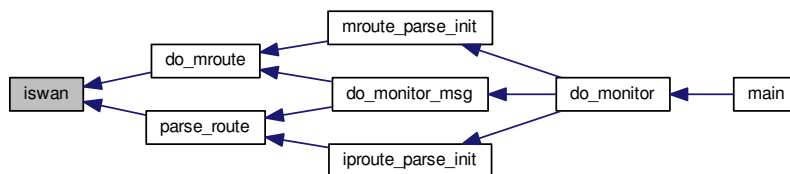
Definition at line 497 of file mcast-pa.c.

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

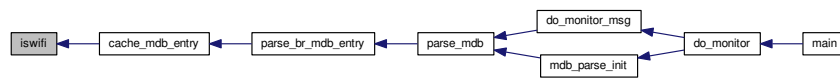
Author

tim.hayes@smartrg.com

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

Referenced by `cache_mdb_entry()`.

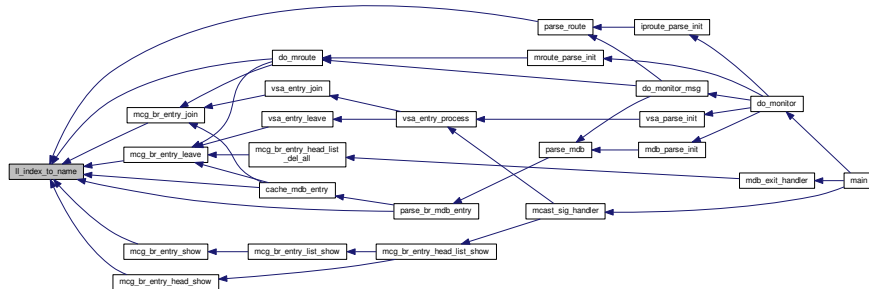
Here is the caller graph for this function:



6.2.3.14 `const char* ll_index_to_name (unsigned idx)`

Referenced by `cache_mdb_entry()`, `do_mroute()`, `mcast_br_entry_head_show()`, `mcast_br_entry_join()`, `mcast_br_entry_leave()`, `mcast_br_entry_show()`, `parse_br_mdb_entry()`, and `parse_route()`.

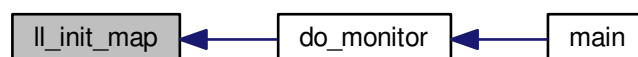
Here is the caller graph for this function:



6.2.3.15 `void ll_init_map (struct rtnl_handle * rth)`

Referenced by `do_monitor()`.

Here is the caller graph for this function:



Returns

pointer to entry or null

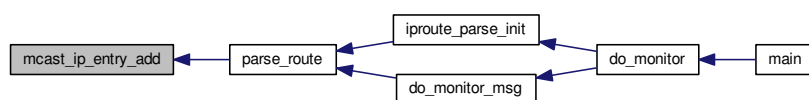
Note**Author**

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

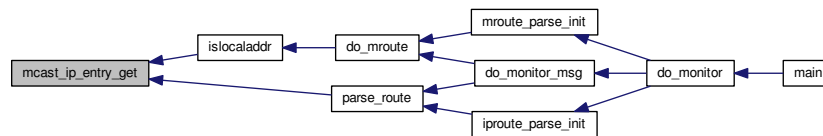
Author

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

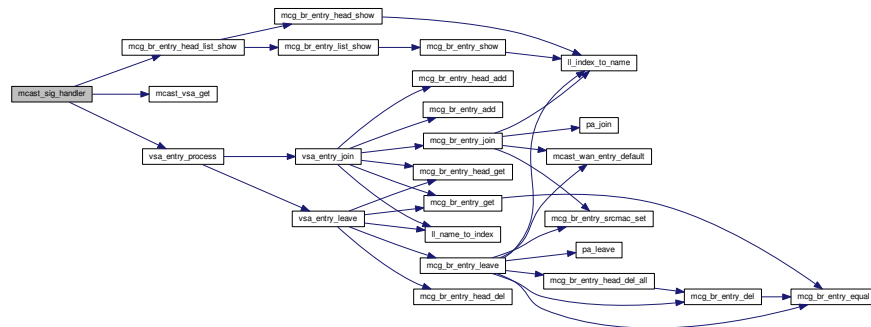
Author

tim.hayes@smartrg.com

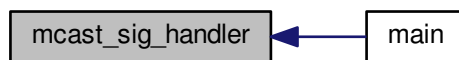
Definition at line 2189 of file mcast-pa.c.

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

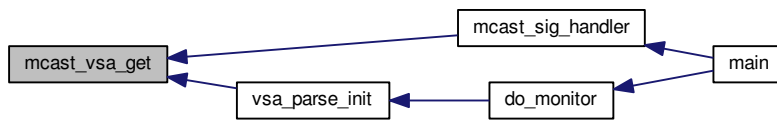
Author

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

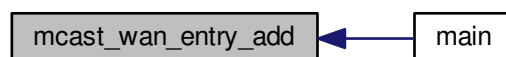
Author

tim.hayes@smartrg.com

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

Referenced by `main()`.

Here is the caller graph for this function:



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

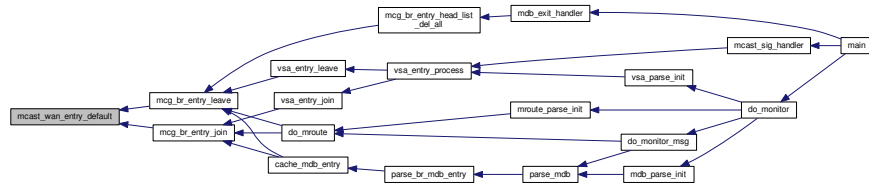
Author

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

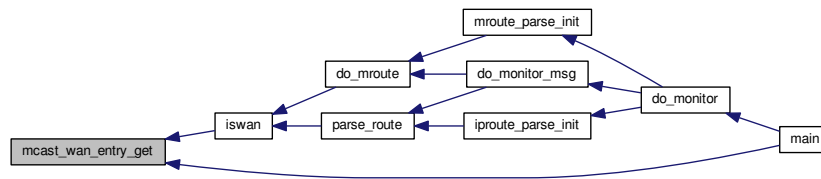
Author

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

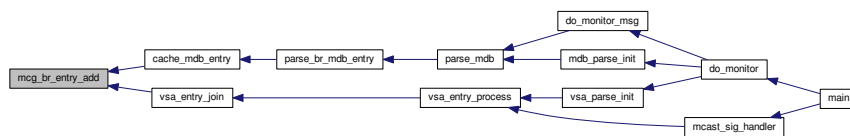
Author

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:



Author

tim.hayes@smartrg.com

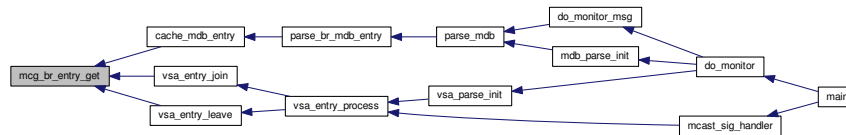
Definition at line 750 of file mcast-pa.c.

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

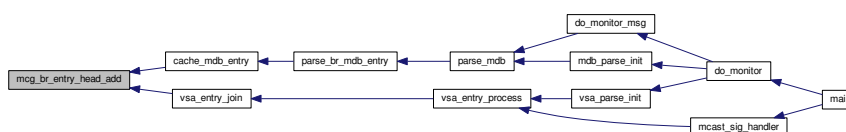
Author

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

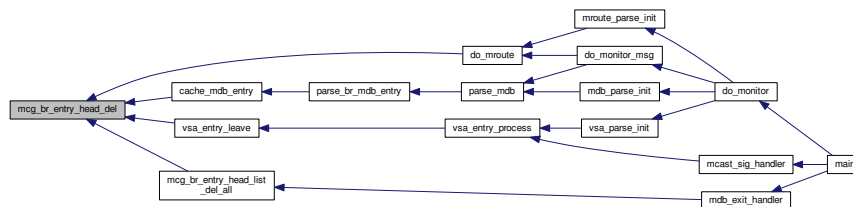
Author

tim.hayes@smartrg.com

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

Referenced by `cache_mdb_entry()`, `do_mroutel()`, `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

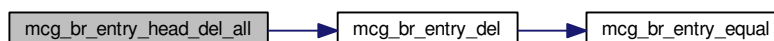
Author

tim.hayes@smartrg.com

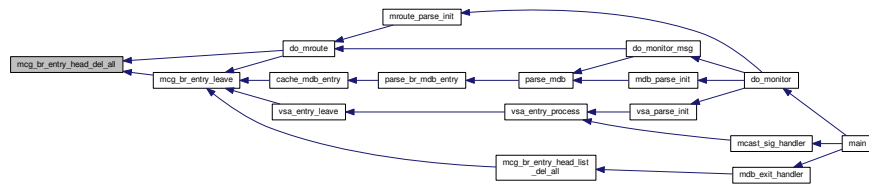
Definition at line 875 of file mcast-pa.c.

Referenced by `do_mroutel()`, 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

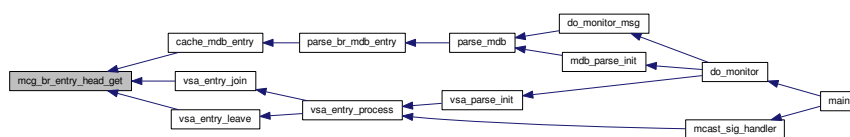
Author

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

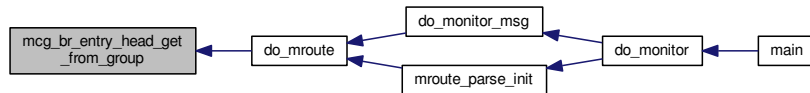
Author

tim.hayes@smartrg.com

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

Referenced by `do_mroutel()`.

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

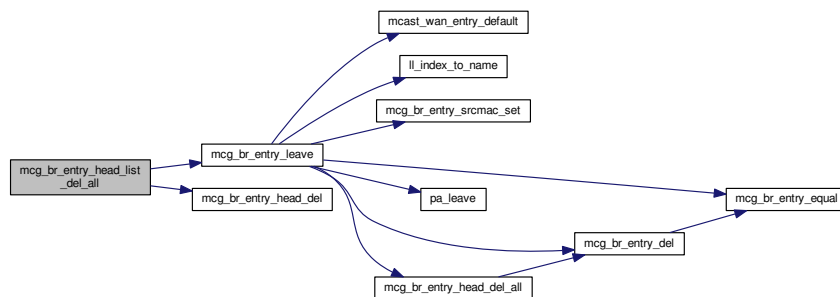
Note**Author**

tim.hayes@smartrg.com

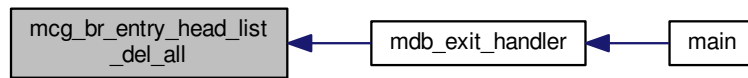
Definition at line 896 of file mcast-pa.c.

Referenced by `mdl_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

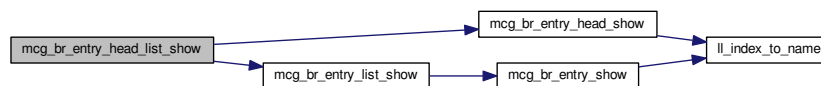
Author

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

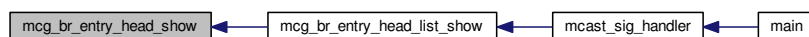
Definition at line 654 of file mcast-pa.c.

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

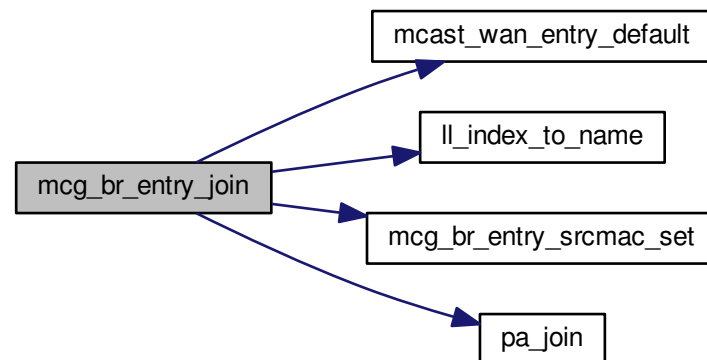
tim.hayes@smartrg.com

< ip address of video source - head use only

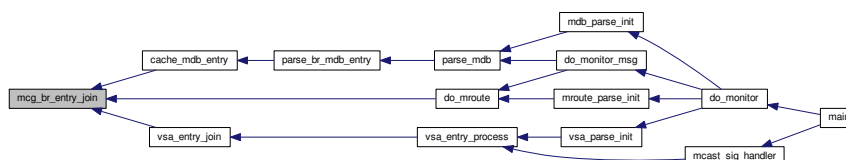
Definition at line 956 of file mcast-pa.c.

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

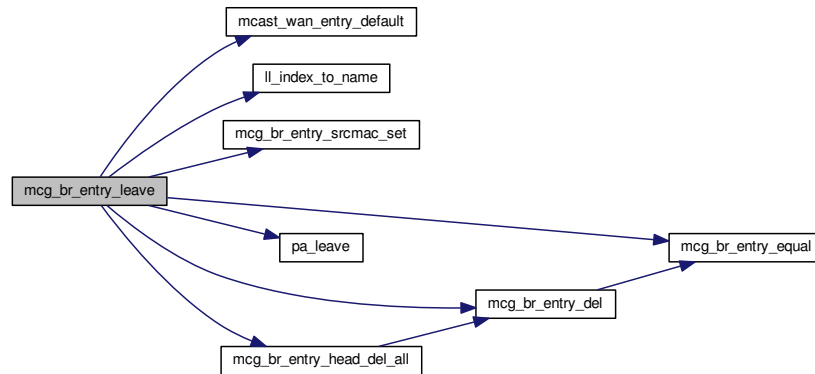
tim.hayes@smartrg.com

< ip address of video source - head use only

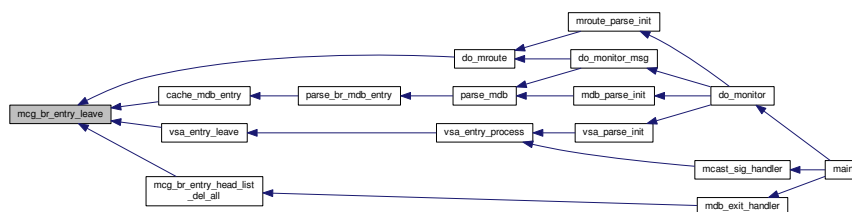
Definition at line 1029 of file mcast-pa.c.

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

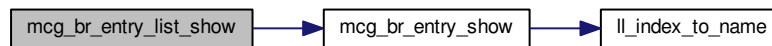
Author

tim.hayes@smartrg.com

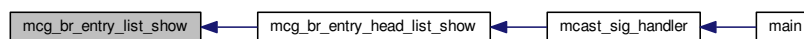
Definition at line 918 of file mcast-pa.c.

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

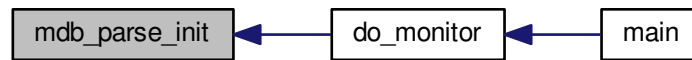
Definition at line 677 of file mcast-pa.c.

Referenced by `mcg_br_entry_list_show()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.3.49 void mroute_bridge_init (void)

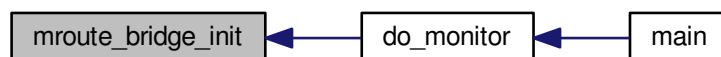
Definition at line 2010 of file mcast-pa.c.

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

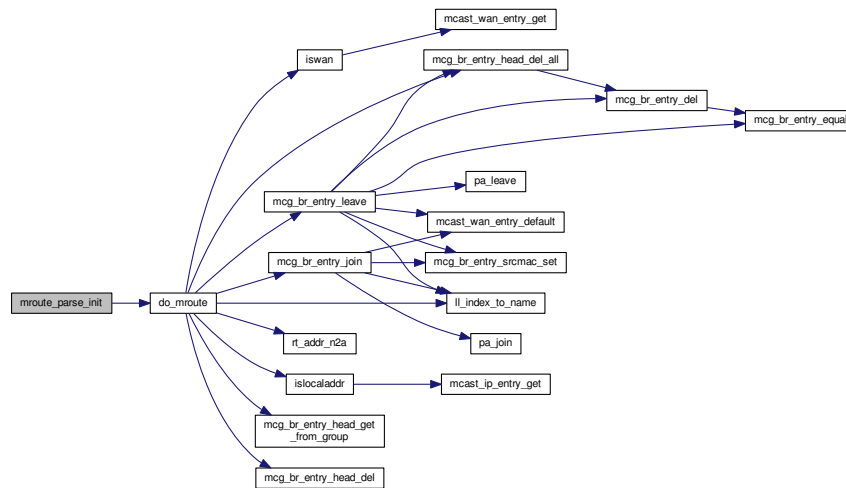
Author

tim.hayes@smartrg.com

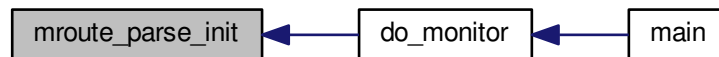
Definition at line 1985 of file mcast-pa.c.

Referenced by `do_monitor()`.

Here is the call graph for this function:



Here is the caller graph for this function:

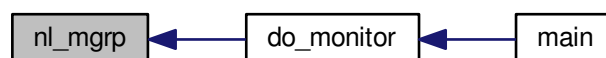


6.2.3.51 `static __u32 nl_mgrp (__u32 group)` `[inline]`, `[static]`

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

Referenced by `do_monitor()`.

Here is the caller graph for this function:

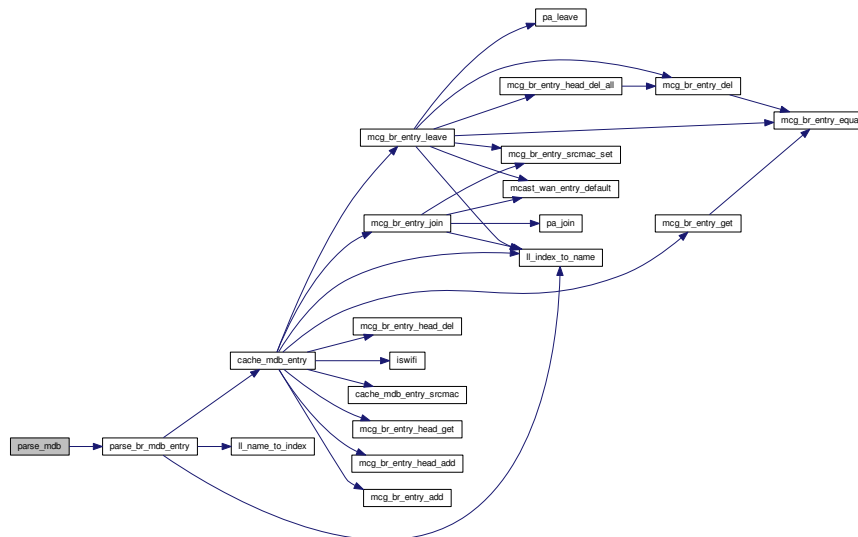


6.2.3.53 `int parse_mdb (const struct sockaddr_nl * who, struct nlmsg_hdr * n, void * arg)`

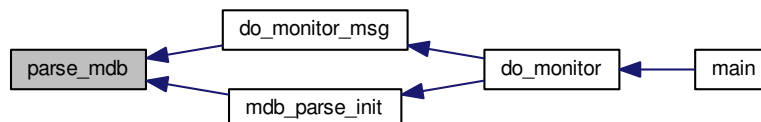
Definition at line 1799 of file mcast-pa.c.

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 nlmsg_hdr * n, void * arg)`

parses route update and gets our host ip address

Todo support for multihoming and review

Note

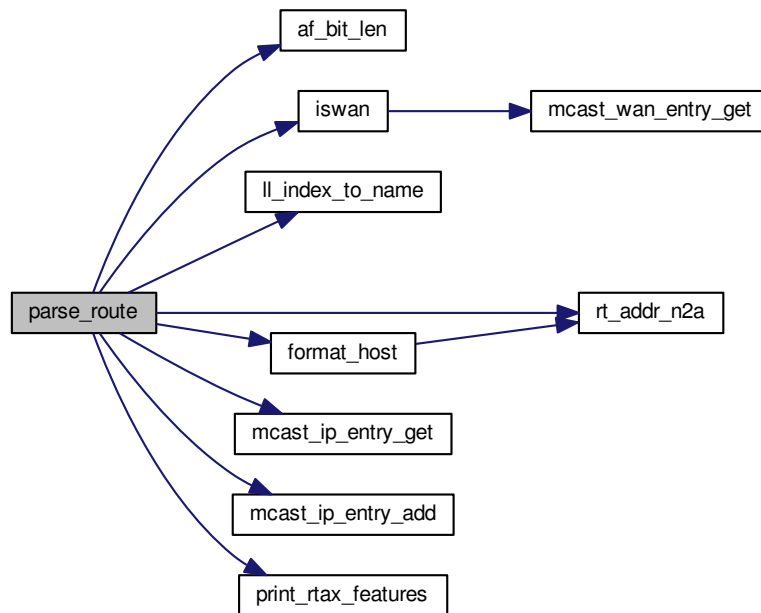
Author

tim.hayes@smartrg.com

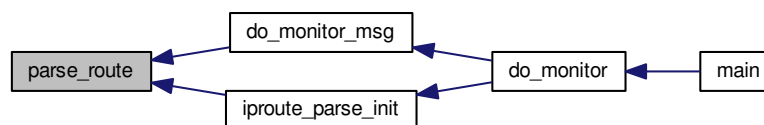
Definition at line 1375 of file mcast-pa.c.

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

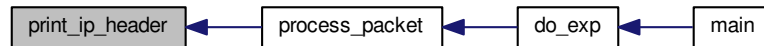
Author

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

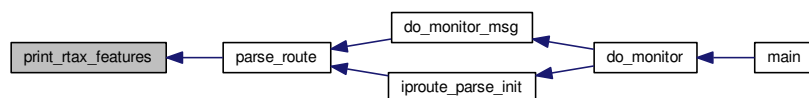
Author

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

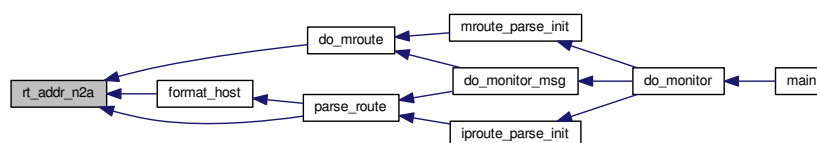
Author

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

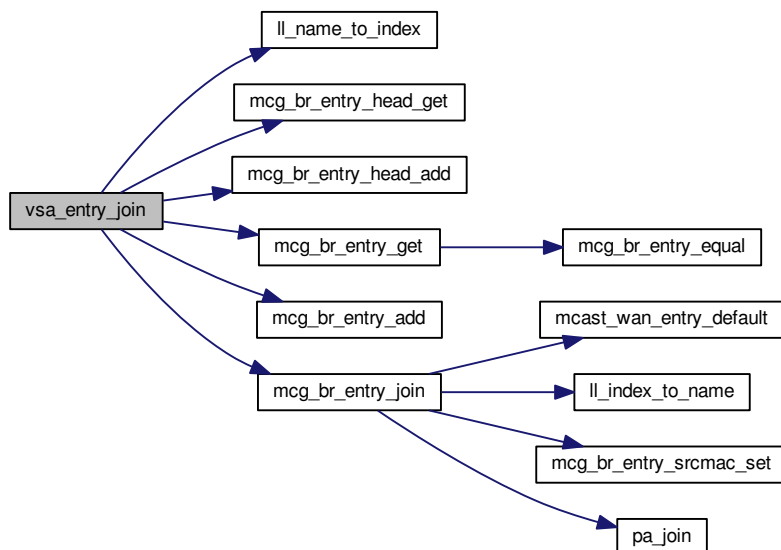
Author

tim.hayes@smartrg.com

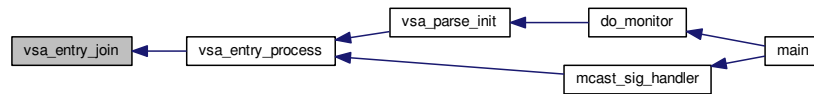
Definition at line 1134 of file mcast-pa.c.

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

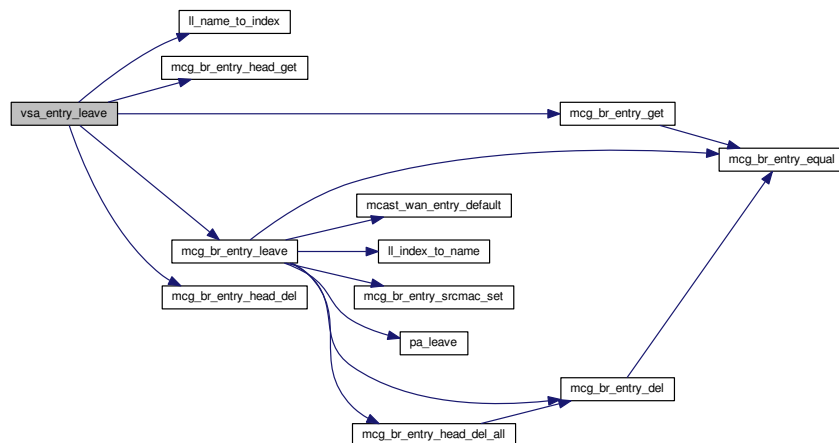
Author

tim.hayes@smartrg.com

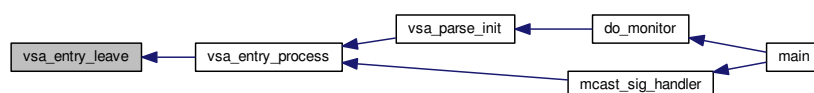
Definition at line 1170 of file mcast-pa.c.

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

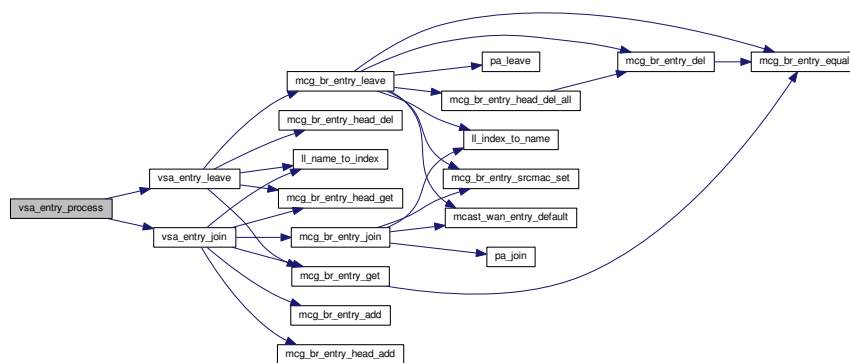
Author

tim.hayes@smartrg.com

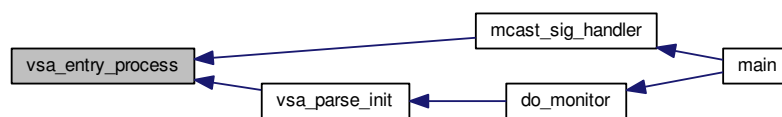
Definition at line 1209 of file mcast-pa.c.

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

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] = "advms",
    [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}

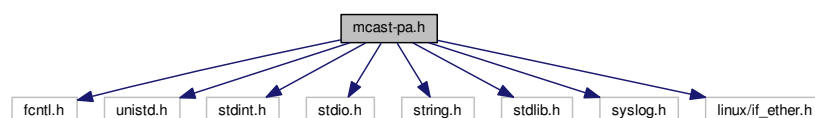
Definition at line 262 of file mcast-pa.c.

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

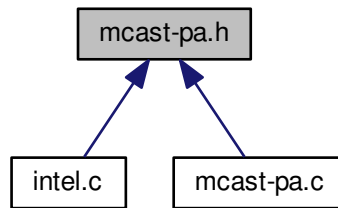
6.3 mcast-pa.h File Reference

```
#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

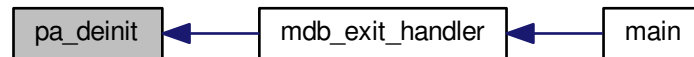
Author

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

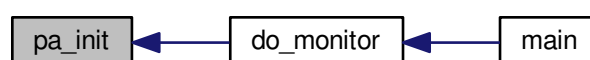
Definition at line 317 of file intel.c.

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

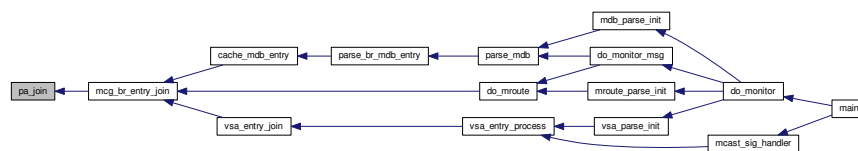
Author

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

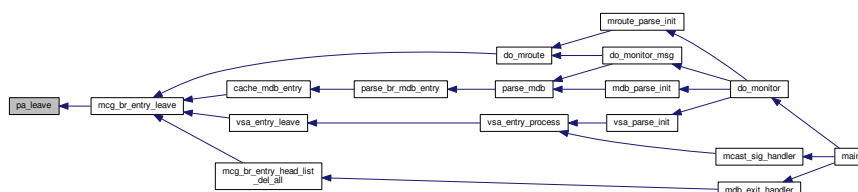
Author

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

- [_SL_](#)
 - [mcast-pa.c, 74](#)
- [address](#)
 - [mcast_ip_entry_t, 11](#)
 - [mcast_wan_entry_t, 12](#)
- [af_bit_len](#)
 - [mcast-pa.c, 33](#)
- [br_ifindex](#)
 - [mcg_br_mdb_entry_t, 18](#)
- [bridge_name](#)
 - [params_t, 20](#)
- [bridged](#)
 - [params_t, 20](#)
- [CMD_BUF_SIZE](#)
 - [mcast-pa.c, 32](#)
- [cache_mdb_entry](#)
 - [mcast-pa.c, 34](#)
- [cache_mdb_entry_srcmac](#)
 - [mcast-pa.c, 34](#)
- [current_time](#)
 - [mcastpa_t, 16](#)
- [dbg](#)
 - [params_t, 20](#)
- [dev_name](#)
 - [mcastpa_t, 16](#)
- [device](#)
 - [vsa_t, 22](#)
- [do_exp](#)
 - [mcast-pa.c, 35](#)
- [do_monitor](#)
 - [mcast-pa.c, 35](#)
- [do_monitor_msg](#)
 - [mcast-pa.c, 36](#)
- [do_mroute](#)
 - [mcast-pa.c, 37](#)
- [do_wait_wan](#)
 - [mcast-pa.c, 38](#)
- [e](#)
 - [mcg_br_mdb_entry_t, 18](#)
- [error_ctime](#)
 - [mcastpa_t, 16](#)
- [exp](#)
 - [params_t, 20](#)
- [fapi_mch_init_sos](#)
 - [intel.c, 24](#)
- [flags](#)
 - [mcastpa_join_leave_t, 13](#)
 - [mcastpa_system_init_t, 14](#)
- [foreground](#)
 - [params_t, 20](#)
- [format_host](#)
 - [mcast-pa.c, 39](#)
- [group](#)
 - [mcastpa_join_leave_t, 13](#)
 - [mcastpa_t, 16](#)
 - [vsa_t, 22](#)
- [head](#)
 - [mcast_ip_entry_t, 11](#)
 - [mcast_wan_entry_t, 12](#)
- [INET_ADDR_SIZE](#)
 - [mcast-pa.c, 32](#)
- [ifindex](#)
 - [mcast_ip_entry_t, 11](#)
 - [mcast_wan_entry_t, 12](#)
- [intel.c, 23](#)
 - [fapi_mch_init_sos, 24](#)
 - [MCAST_HELPER_DEVICE, 24](#)
 - [pa_deinit, 25](#)
 - [pa_init, 25](#)
 - [pa_init_sos, 26](#)
 - [pa_join, 27](#)
 - [pa_leave, 27](#)
- [ip_head](#)
 - [mcastpa_t, 16](#)
- [iproute_parse_init](#)
 - [mcast-pa.c, 40](#)
- [islocaladdr](#)
 - [mcast-pa.c, 41](#)
- [iswan](#)
 - [mcast-pa.c, 41](#)
- [iswifi](#)
 - [mcast-pa.c, 42](#)
- [joined](#)
 - [mcg_br_mdb_entry_t, 18](#)
- [lan](#)
 - [mcastpa_join_leave_t, 13](#)
- [lan_dev](#)
 - [mcastpa_join_leave_t, 13](#)
- [last_time](#)
 - [mcastpa_t, 16](#)
- [ll_index_to_name](#)

- mcast-pa.c, [43](#)
- ll_init_map
 - mcast-pa.c, [43](#)
- ll_name_to_index
 - mcast-pa.c, [43](#)
- long_options
 - mcast-pa.c, [74](#)
- MCAST_HELPER_DEVICE
 - intel.c, [24](#)
- MCASTPA_STRING_SIZE
 - mcast-pa.h, [76](#)
- MDB_FORK_EXIT
 - mcast-pa.c, [32](#)
- MDBA_RTAA
 - mcast-pa.c, [32](#)
- MJL_FLAG_BRIDGE
 - mcast-pa.h, [76](#)
- MJL_FLAG_EXP
 - mcast-pa.h, [77](#)
- MJL_FLAG_LAN
 - mcast-pa.h, [77](#)
- MJL_FLAG_SRCIP
 - mcast-pa.h, [77](#)
- MJL_FLAG_UPDATE
 - mcast-pa.h, [77](#)
- MROUTE_DEFAULT_TTL
 - mcast-pa.c, [32](#)
- MSI_FLAG_EXP
 - mcast-pa.h, [77](#)
- main
 - mcast-pa.c, [44](#)
- mcast-pa.c, [28](#)
 - _SL_, [74](#)
 - af_bit_len, [33](#)
 - CMD_BUF_SIZE, [32](#)
 - cache_mdb_entry, [34](#)
 - cache_mdb_entry_srcmac, [34](#)
 - do_exp, [35](#)
 - do_monitor, [35](#)
 - do_monitor_msg, [36](#)
 - do_mroute, [37](#)
 - do_wait_wan, [38](#)
 - format_host, [39](#)
 - INET_ADDR_SIZE, [32](#)
 - iproute_parse_init, [40](#)
 - islocaladdr, [41](#)
 - iswan, [41](#)
 - iswifi, [42](#)
 - ll_index_to_name, [43](#)
 - ll_init_map, [43](#)
 - ll_name_to_index, [43](#)
 - long_options, [74](#)
 - MDB_FORK_EXIT, [32](#)
 - MDBA_RTAA, [32](#)
 - MROUTE_DEFAULT_TTL, [32](#)
 - main, [44](#)
 - mcast_ip_entry_add, [44](#)
 - mcast_ip_entry_del, [45](#)
 - mcast_ip_entry_get, [45](#)
 - mcast_ip_entry_list, [46](#)
 - mcast_sig_handler, [46](#)
 - mcast_vsa_get, [47](#)
 - mcast_wan_entry_add, [48](#)
 - mcast_wan_entry_default, [48](#)
 - mcast_wan_entry_del, [49](#)
 - mcast_wan_entry_get, [49](#)
 - mcast_wan_entry_list, [50](#)
 - mcastpa, [74](#)
 - mcastpa_usage, [50](#)
 - mcg_br_entry_add, [50](#)
 - mcg_br_entry_del, [51](#)
 - mcg_br_entry_equal, [52](#)
 - mcg_br_entry_get, [52](#)
 - mcg_br_entry_head_add, [53](#)
 - mcg_br_entry_head_del, [54](#)
 - mcg_br_entry_head_del_all, [54](#)
 - mcg_br_entry_head_get, [55](#)
 - mcg_br_entry_head_get_from_group, [55](#)
 - mcg_br_entry_head_list_del_all, [56](#)
 - mcg_br_entry_head_list_show, [57](#)
 - mcg_br_entry_head_show, [57](#)
 - mcg_br_entry_join, [58](#)
 - mcg_br_entry_leave, [59](#)
 - mcg_br_entry_list_show, [60](#)
 - mcg_br_entry_show, [61](#)
 - mcg_br_entry_srcmac_set, [62](#)
 - mdb_exit_handler, [62](#)
 - mdb_parse_init, [63](#)
 - mroute_bridge_init, [64](#)
 - mroute_parse_init, [64](#)
 - mx_names, [75](#)
 - nl_mgrp, [65](#)
 - PRTFL, [33](#)
 - parse_br_mdb_entry, [66](#)
 - parse_mdb, [66](#)
 - parse_route, [67](#)
 - print_ip_header, [68](#)
 - print_rtax_features, [69](#)
 - process_packet, [69](#)
 - rt_addr_n2a, [70](#)
 - rth, [75](#)
 - rtm_get_table, [70](#)
 - SPRINT_BSIZE, [33](#)
 - SPRINT_BUF, [33](#)
 - vsa_entry_join, [71](#)
 - vsa_entry_leave, [72](#)
 - vsa_entry_process, [72](#)
 - vsa_parse_init, [73](#)
- mcast-pa.h, [75](#)
 - MCASTPA_STRING_SIZE, [76](#)
 - MJL_FLAG_BRIDGE, [76](#)
 - MJL_FLAG_EXP, [77](#)
 - MJL_FLAG_LAN, [77](#)
 - MJL_FLAG_SRCIP, [77](#)
 - MJL_FLAG_UPDATE, [77](#)
 - MSI_FLAG_EXP, [77](#)

- pa_deinit, 77
- pa_init, 78
- pa_join, 78
- pa_leave, 79
- mcast_ip_entry_add
 - mcast-pa.c, 44
- mcast_ip_entry_del
 - mcast-pa.c, 45
- mcast_ip_entry_get
 - mcast-pa.c, 45
- mcast_ip_entry_list
 - mcast-pa.c, 46
- mcast_ip_entry_t, 11
 - address, 11
 - head, 11
 - ifindex, 11
 - name, 11
- mcast_sig_handler
 - mcast-pa.c, 46
- mcast_vsa_get
 - mcast-pa.c, 47
- mcast_wan_entry_add
 - mcast-pa.c, 48
- mcast_wan_entry_default
 - mcast-pa.c, 48
- mcast_wan_entry_del
 - mcast-pa.c, 49
- mcast_wan_entry_get
 - mcast-pa.c, 49
- mcast_wan_entry_list
 - mcast-pa.c, 50
- mcast_wan_entry_t, 12
 - address, 12
 - head, 12
 - ifindex, 12
 - name, 12
- mcastpa
 - mcast-pa.c, 74
- mcastpa_join_leave_t, 13
 - flags, 13
 - group, 13
 - lan, 13
 - lan_dev, 13
 - srcip, 13
 - srcmac, 14
 - wan, 14
- mcastpa_system_init_t, 14
 - flags, 14
 - srcip, 14
 - wan, 14
- mcastpa_t, 15
 - current_time, 16
 - dev_name, 16
 - error_ctime, 16
 - group, 16
 - ip_head, 16
 - last_time, 16
 - mcg_head, 16
 - params, 16
 - sock, 17
 - start_ctime, 17
 - start_time, 17
 - timer_tick, 17
 - wan_head, 17
- mcastpa_usage
 - mcast-pa.c, 50
- mcg_br_entry_add
 - mcast-pa.c, 50
- mcg_br_entry_del
 - mcast-pa.c, 51
- mcg_br_entry_equal
 - mcast-pa.c, 52
- mcg_br_entry_get
 - mcast-pa.c, 52
- mcg_br_entry_head_add
 - mcast-pa.c, 53
- mcg_br_entry_head_del
 - mcast-pa.c, 54
- mcg_br_entry_head_del_all
 - mcast-pa.c, 54
- mcg_br_entry_head_get
 - mcast-pa.c, 55
- mcg_br_entry_head_get_from_group
 - mcast-pa.c, 55
- mcg_br_entry_head_list_del_all
 - mcast-pa.c, 56
- mcg_br_entry_head_list_show
 - mcast-pa.c, 57
- mcg_br_entry_head_show
 - mcast-pa.c, 57
- mcg_br_entry_join
 - mcast-pa.c, 58
- mcg_br_entry_leave
 - mcast-pa.c, 59
- mcg_br_entry_list_show
 - mcast-pa.c, 60
- mcg_br_entry_show
 - mcast-pa.c, 61
- mcg_br_entry_srcmac_set
 - mcast-pa.c, 62
- mcg_br_mdb_entry_t, 17
 - br_ifindex, 18
 - e, 18
 - joined, 18
 - mcg_entry, 18
 - mcg_head, 18
 - src, 18
 - wan_ifindex, 18
- mcg_entry
 - mcg_br_mdb_entry_t, 18
- mcg_head
 - mcastpa_t, 16
 - mcg_br_mdb_entry_t, 18
- mdb_exit_handler
 - mcast-pa.c, 62
- mdb_parse_init

- mcast-pa.c, 63
- monitor
 - params_t, 20
- mroute_bridge_init
 - mcast-pa.c, 64
- mroute_parse_init
 - mcast-pa.c, 64
- mx_names
 - mcast-pa.c, 75
- name
 - mcast_ip_entry_t, 11
 - mcast_wan_entry_t, 12
- nl_mgrp
 - mcast-pa.c, 65
- nowifi
 - params_t, 20
- op
 - vsa_t, 22
- PRTFL
 - mcast-pa.c, 33
- pa_deinit
 - intel.c, 25
 - mcast-pa.h, 77
- pa_init
 - intel.c, 25
 - mcast-pa.h, 78
- pa_init_sos
 - intel.c, 26
- pa_join
 - intel.c, 27
 - mcast-pa.h, 78
- pa_leave
 - intel.c, 27
 - mcast-pa.h, 79
- params
 - mcastpa_t, 16
- params_t, 19
 - bridge_name, 20
 - bridged, 20
 - dbg, 20
 - exp, 20
 - foreground, 20
 - monitor, 20
 - nowifi, 20
 - src, 20
 - use_src, 21
 - verbose, 21
 - video2lan, 21
 - video2lan_name, 21
 - vsa, 21
 - wan, 21
 - wan_ifindex, 21
- parse_br_mdb_entry
 - mcast-pa.c, 66
- parse_mdb
 - mcast-pa.c, 66
- parse_route
 - mcast-pa.c, 67
- print_ip_header
 - mcast-pa.c, 68
- print_rtax_features
 - mcast-pa.c, 69
- process_packet
 - mcast-pa.c, 69
- rt_addr_n2a
 - mcast-pa.c, 70
- rth
 - mcast-pa.c, 75
- rtm_get_table
 - mcast-pa.c, 70
- SPRINT_BSIZE
 - mcast-pa.c, 33
- SPRINT_BUF
 - mcast-pa.c, 33
- sock
 - mcastpa_t, 17
- src
 - mcbg_br_mdb_entry_t, 18
 - params_t, 20
- srcip
 - mcastpa_join_leave_t, 13
 - mcastpa_system_init_t, 14
- srcmac
 - mcastpa_join_leave_t, 14
- start_ctime
 - mcastpa_t, 17
- start_time
 - mcastpa_t, 17
- timer_tick
 - mcastpa_t, 17
- use_src
 - params_t, 21
- valid
 - vsa_t, 22
- verbose
 - params_t, 21
- video2lan
 - params_t, 21
- video2lan_name
 - params_t, 21
- vsa
 - params_t, 21
- vsa_entry_join
 - mcast-pa.c, 71
- vsa_entry_leave
 - mcast-pa.c, 72
- vsa_entry_process
 - mcast-pa.c, 72
- vsa_parse_init
 - mcast-pa.c, 73

- vsa_t, [22](#)
 - device, [22](#)
 - group, [22](#)
 - op, [22](#)
 - valid, [22](#)
- wan
 - mcastpa_join_leave_t, [14](#)
 - mcastpa_system_init_t, [14](#)
 - params_t, [21](#)
- wan_head
 - mcastpa_t, [17](#)
- wan_ifindex
 - mcbg_br_mdb_entry_t, [18](#)
 - params_t, [21](#)