

sonata

Generated by Doxygen 1.5.6

Tue Jun 17 17:52:03 2008

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	sctp_nat_assoc Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Member Function Documentation	6
3.1.2.1	LIST_ENTRY	6
3.1.2.2	LIST_ENTRY	6
3.1.2.3	LIST_ENTRY	6
3.1.3	Field Documentation	6
3.1.3.1	l_vtag	6
3.1.3.2	l_port	6
3.1.3.3	g_vtag	6
3.1.3.4	g_port	6
3.1.3.5	l_addr	7
3.1.3.6	state	7
3.1.3.7	TableRegister	7
3.1.3.8	exp	7
3.1.3.9	exp_loc	7
3.2	sctp_nat_msg Struct Reference	8
3.2.1	Detailed Description	8
3.2.2	Field Documentation	8
3.2.2.1	msg	8
3.2.2.2	ip_hdr	8
3.2.2.3	sctp_hdr	8

3.2.2.4	sctpchnk	9
3.3	sctp_nat_timer Struct Reference	10
3.3.1	Detailed Description	10
3.3.2	Member Function Documentation	10
3.3.2.1	LIST_HEAD	10
3.3.3	Field Documentation	10
3.3.3.1	loc_time	10
3.3.3.2	cur_loc	10
3.4	sctpChunkOfInt Union Reference	11
3.4.1	Detailed Description	11
3.4.2	Field Documentation	11
3.4.2.1	Init	11
3.4.2.2	InitAck	11
3.4.2.3	Asconf	11
4	File Documentation	13
4.1	alias_sctp.c File Reference	13
4.1.1	Detailed Description	16
4.1.2	Define Documentation	16
4.1.2.1	sn_calloc	16
4.1.2.2	sn_free	16
4.1.2.3	sn_malloc	17
4.1.3	Function Documentation	17
4.1.3.1	AddSctpAssocGlobal	17
4.1.3.2	AddSctpAssocLocal	17
4.1.3.3	AliasSctpInit	18
4.1.3.4	AliasSctpTerm	18
4.1.3.5	CL_process	19
4.1.3.6	FindSctpGlobal	19
4.1.3.7	FindSctpGlobalT	20
4.1.3.8	FindSctpLocal	20
4.1.3.9	FindSctpLocalT	21
4.1.3.10	ID_process	22
4.1.3.11	INa_process	22
4.1.3.12	INi_process	23
4.1.3.13	logsectpassoc	24
4.1.3.14	logsectperror	24

4.1.3.15	logSctpGlobal	25
4.1.3.16	logSctpLocal	25
4.1.3.17	logscpparse	26
4.1.3.18	logTimerQ	26
4.1.3.19	ProcessSctpMsg	27
4.1.3.20	ReplyAbortM	27
4.1.3.21	ReplyErrorM	28
4.1.3.22	RmSctpAssoc	28
4.1.3.23	sctp_AddTimeOut	29
4.1.3.24	sctp_CheckExp	29
4.1.3.25	sctp_FirstChunkHdr	30
4.1.3.26	sctp_NextChunkHdr	30
4.1.3.27	SCTP_PktParser	30
4.1.3.28	sctp_ResetTimeOut	31
4.1.3.29	sctp_RmTimeOut	32
4.1.3.30	SctpAliasIn	32
4.1.3.31	SctpAliasLog	33
4.1.3.32	SctpAliasOut	33
4.1.3.33	SctpShowAliasStats	35
4.1.3.34	StartPointGlobal	35
4.1.3.35	StartPointLocal	35
4.1.3.36	UP_process	35
4.2	alias_sctp.h File Reference	37
4.2.1	Detailed Description	39
4.2.2	Define Documentation	39
4.2.2.1	_SCTP_NAT_DEBUG	39
4.2.2.2	LINK_SCTP	39
4.2.2.3	MAX_SCTP_TIMERQ	39
4.2.2.4	SCTP_PACKED	40
4.2.2.5	SCTP_UNUSED	40
4.2.2.6	sctpAbort	40
4.2.2.7	sctpAddIp	40
4.2.2.8	sctpAddIpAck	40
4.2.2.9	sctpInit	40
4.2.2.10	sctpInitAck	40
4.2.2.11	sctpOther	41

4.2.2.12	sctpShutAck	41
4.2.2.13	sctpShutComp	41
4.2.2.14	SN_A_T	41
4.2.2.15	SN_ADD_CLASH	41
4.2.2.16	SN_ADD_OK	41
4.2.2.17	SN_BOTH_TBL	41
4.2.2.18	SN_C_T	42
4.2.2.19	SN_CL	42
4.2.2.20	SN_CURTIME	42
4.2.2.21	SN_DROP_PKT	42
4.2.2.22	SN_GLOBAL_TBL	42
4.2.2.23	SN_I_T	42
4.2.2.24	SN_ID	42
4.2.2.25	SN_INa	43
4.2.2.26	SN_INi	43
4.2.2.27	SN_LOCAL_TBL	43
4.2.2.28	SN_NAT_PKT	43
4.2.2.29	SN_NULL_TBL	43
4.2.2.30	SN_PARSE_ERROR_AS_MALLOC	43
4.2.2.31	SN_PARSE_ERROR_CHHL	43
4.2.2.32	SN_PARSE_ERROR_CHUNK	44
4.2.2.33	SN_PARSE_ERROR_DIR	44
4.2.2.34	SN_PARSE_ERROR_IPSHL	44
4.2.2.35	SN_PARSE_ERROR_VTAG	44
4.2.2.36	SN_PARSE_OK	44
4.2.2.37	SN_PROCESSING_ERROR	44
4.2.2.38	SN_REPLY_ABORT	44
4.2.2.39	SN_REPLY_ERROR	45
4.2.2.40	SN_RM	45
4.2.2.41	SN_SEND_ABORT	45
4.2.2.42	SN_SHUTDOWN	45
4.2.2.43	SN_TABLE_GLOBAL_SIZE	45
4.2.2.44	SN_TABLE_LOCAL_SIZE	45
4.2.2.45	SN_TO_GLOBAL	45
4.2.2.46	SN_TO_LOCAL	46
4.2.2.47	SN_U_T	46

4.2.2.48	SN_UP	46
4.2.2.49	SN_UT	46
4.2.2.50	SN_X_T	46

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

sctp_nat_assoc (Sctp association information)	5
sctp_nat_msg (SCTP message)	8
sctp_nat_timer (Sctp nat timer queue structure)	10
sctpChunkOfInt (SCTP chunk of interest)	11

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

alias_sctp.c	13
alias_sctp.h	37

Chapter 3

Data Structure Documentation

3.1 sctp_nat_assoc Struct Reference

sctp association information

```
#include <alias_sctp.h>
```

Public Member Functions

- [LIST_ENTRY \(sctp_nat_assoc\) list_L](#)
- [LIST_ENTRY \(sctp_nat_assoc\) list_G](#)
- [LIST_ENTRY \(sctp_nat_assoc\) timer_Q](#)

Data Fields

- [uint32_t l_vtag](#)
- [uint16_t l_port](#)
- [uint32_t g_vtag](#)
- [uint16_t g_port](#)
- [struct in_addr l_addr](#)
- [int state](#)
- [int TableRegister](#)
- [int exp](#)
- [int exp_loc](#)

3.1.1 Detailed Description

sctp association information

Structure that contains information about a particular sctp association currently under Network Address Translation. Information is stored in network byte order (as is libalias)***

Definition at line 205 of file alias_sctp.h.

3.1.2 Member Function Documentation

3.1.2.1 `sctp_nat_assoc::LIST_ENTRY (sctp_nat_assoc)`

Linked list of pointers for Local table

3.1.2.2 `sctp_nat_assoc::LIST_ENTRY (sctp_nat_assoc)`

Linked list of pointers for Global table

3.1.2.3 `sctp_nat_assoc::LIST_ENTRY (sctp_nat_assoc)`

Linked list of pointers for timer Q

3.1.3 Field Documentation

3.1.3.1 `uint32_t sctp_nat_assoc::l_vtag`

local side verification tag

Definition at line 206 of file `alias_sctp.h`.

Referenced by `AddSctpAssocLocal()`, `FindSctpGlobalT()`, `FindSctpLocal()`, `ID_process()`, `INi_process()`, and `logscpassoc()`.

3.1.3.2 `uint16_t sctp_nat_assoc::l_port`

local side port number

Definition at line 207 of file `alias_sctp.h`.

Referenced by `AddSctpAssocGlobal()`, `AddSctpAssocLocal()`, `FindSctpGlobalT()`, `FindSctpLocal()`, `ID_process()`, and `logscpassoc()`.

3.1.3.3 `uint32_t sctp_nat_assoc::g_vtag`

global side verification tag

Definition at line 208 of file `alias_sctp.h`.

Referenced by `AddSctpAssocGlobal()`, `FindSctpGlobal()`, `FindSctpLocalT()`, `ID_process()`, `INi_process()`, `logscpassoc()`, and `sctp_CheckExp()`.

3.1.3.4 `uint16_t sctp_nat_assoc::g_port`

global side port number

Definition at line 209 of file `alias_sctp.h`.

Referenced by `AddSctpAssocGlobal()`, `AddSctpAssocLocal()`, `FindSctpGlobal()`, `FindSctpLocalT()`, `ID_process()`, and `logscpassoc()`.

3.1.3.5 struct in_addr sctp_nat_assoc::l_addr [read]

local ip address

Definition at line 210 of file alias_sctp.h.

Referenced by AddSctpAssocGlobal(), AddSctpAssocLocal(), FindSctpLocal(), ID_process(), INi_process(), logsctpassoc(), and SctpAliasIn().

3.1.3.6 int sctp_nat_assoc::state

current state of NAT association

Definition at line 211 of file alias_sctp.h.

Referenced by CL_process(), ID_process(), INa_process(), INi_process(), logsctpassoc(), ProcessSctpMsg(), sctp_CheckExp(), SctpAliasIn(), SctpAliasOut(), and UP_process().

3.1.3.7 int sctp_nat_assoc::TableRegister

stores which look up tables association is registered in

Definition at line 212 of file alias_sctp.h.

Referenced by AddSctpAssocGlobal(), AddSctpAssocLocal(), logsctpassoc(), RmSctpAssoc(), SctpAliasIn(), and SctpAliasOut().

3.1.3.8 int sctp_nat_assoc::exp

timer expiration in seconds from uptime

Definition at line 213 of file alias_sctp.h.

Referenced by ID_process(), logsctpassoc(), sctp_AddTimeOut(), sctp_CheckExp(), and sctp_ResetTimeOut().

3.1.3.9 int sctp_nat_assoc::exp_loc

current location in timer_Q

Definition at line 214 of file alias_sctp.h.

Referenced by sctp_AddTimeOut().

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

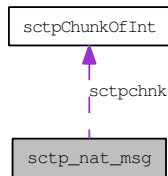
- [alias_sctp.h](#)

3.2 sctp_nat_msg Struct Reference

SCTP message.

```
#include <alias_sctp.h>
```

Collaboration diagram for sctp_nat_msg:



Data Fields

- `uint16_t msg`
- `struct ip * ip_hdr`
- `struct sctphdr * sctp_hdr`
- `union sctpChunkOfInt sctpchnk`

3.2.1 Detailed Description

SCTP message.

Structure containing the relevant information from the SCTP message

Definition at line 239 of file `alias_sctp.h`.

3.2.2 Field Documentation

3.2.2.1 `uint16_t sctp_nat_msg::msg`

one of the key messages defined above

Definition at line 240 of file `alias_sctp.h`.

Referenced by `CL_process()`, `ID_process()`, `INa_process()`, `INi_process()`, `logscpparse()`, `SCTP_PktParser()`, and `UP_process()`.

3.2.2.2 `struct ip* sctp_nat_msg::ip_hdr` [read]

pointer to ip packet header

Definition at line 244 of file `alias_sctp.h`.

Referenced by `ID_process()`, `INi_process()`, `SCTP_PktParser()`, `SctpAliasIn()`, and `SctpAliasOut()`.

3.2.2.3 `struct sctphdr* sctp_nat_msg::sctp_hdr` [read]

pointer to sctp common header

Definition at line 246 of file alias_sctp.h.

Referenced by ID_process(), SCTP_PktParser(), SctpAliasIn(), and SctpAliasOut().

3.2.2.4 union sctpChunkOfInt sctp_nat_msg::sctpchnk [write]

union of pointers to the chunk of interest

Definition at line 247 of file alias_sctp.h.

Referenced by ID_process(), INi_process(), SCTP_PktParser(), SctpAliasIn(), and SctpAliasOut().

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

- [alias_sctp.h](#)

3.3 sctp_nat_timer Struct Reference

sctp nat timer queue structure

```
#include <alias_sctp.h>
```

Public Member Functions

- [LIST_HEAD](#) (sctpTimerQ, sctp_nat_assoc)*TimerQ

Data Fields

- int [loc_time](#)
- int [cur_loc](#)

3.3.1 Detailed Description

sctp nat timer queue structure

Definition at line 258 of file alias_sctp.h.

3.3.2 Member Function Documentation

3.3.2.1 sctp_nat_timer::LIST_HEAD (sctpTimerQ, sctp_nat_assoc)

List of associations at this position in the timer Q

3.3.3 Field Documentation

3.3.3.1 int sctp_nat_timer::loc_time

time in seconds for the current location in the queue

Definition at line 259 of file alias_sctp.h.

3.3.3.2 int sctp_nat_timer::cur_loc

index of the current location in the circular queue

Definition at line 260 of file alias_sctp.h.

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

- [alias_sctp.h](#)

3.4 sctpChunkOfInt Union Reference

SCTP chunk of interest.

```
#include <alias_sctp.h>
```

Data Fields

- struct sctp_init * [Init](#)
- struct sctp_init_ack * [InitAck](#)
- struct sctp_asconf_paramhdr * [Asconf](#)

3.4.1 Detailed Description

SCTP chunk of interest.

The only chunks whose contents are of any interest are the INIT and ASCONF_AddIP

Definition at line 227 of file alias_sctp.h.

3.4.2 Field Documentation

3.4.2.1 struct sctp_init* sctpChunkOfInt::Init [read]

Pointer to Init Chunk

Definition at line 228 of file alias_sctp.h.

Referenced by ID_process(), INi_process(), SCTP_PktParser(), SctpAliasIn(), and SctpAliasOut().

3.4.2.2 struct sctp_init_ack* sctpChunkOfInt::InitAck [read]

Pointer to Init Chunk

Definition at line 229 of file alias_sctp.h.

Referenced by SCTP_PktParser().

3.4.2.3 struct sctp_asconf_paramhdr* sctpChunkOfInt::Asconf [read]

Pointer to ASCONF chunk

Definition at line 230 of file alias_sctp.h.

Referenced by SCTP_PktParser().

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

- [alias_sctp.h](#)

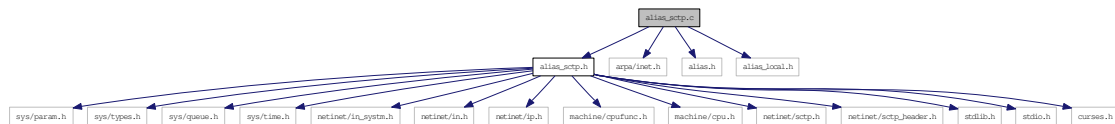
Chapter 4

File Documentation

4.1 alias_sctp.c File Reference

```
#include "alias_sctp.h"  
#include <arpa/inet.h>  
#include "alias.h"  
#include "alias_local.h"
```

Include dependency graph for alias_sctp.c:



Defines

- #define `sn_malloc(x)` `malloc(x)`
- #define `sn_calloc(n, x)` `calloc(n,x)`
- #define `sn_free(x)` `free(x)`

Functions

- static int `SCTP_PktParser` (struct libalias *la, int direction, struct ip *pip, struct `sctp_nat_msg` *sm, struct `sctp_nat_assoc` **passoc)
parses SCTP packets for the key SCTP chunk that will be processed
- static int `ProcessSctpMsg` (struct libalias *la, int direction, struct `sctp_nat_msg` *sm, struct `sctp_nat_assoc` *assoc)
Process SCTP message.
- static int `ID_process` (struct libalias *la, int direction, struct `sctp_nat_assoc` *assoc, struct `sctp_nat_msg` *sm)
Process sctp message while in the Idle state.

- static int [INi_process](#) (struct libalias *la, int direction, struct [sctp_nat_assoc](#) *assoc, struct [sctp_nat_msg](#) *sm)
Process sctp message while waiting for an InitAck message.
- static int [INa_process](#) (struct libalias *la, int direction, struct [sctp_nat_assoc](#) *assoc, struct [sctp_nat_msg](#) *sm)
Process sctp message while waiting for an AddIpAck message.
- static int [UP_process](#) (struct libalias *la, int direction, struct [sctp_nat_assoc](#) *assoc, struct [sctp_nat_msg](#) *sm)
Process sctp messages while association is UP redirecting packets.
- static int [CL_process](#) (struct libalias *la, int direction, struct [sctp_nat_assoc](#) *assoc, struct [sctp_nat_msg](#) *sm)
Process sctp message while association is in the process of closing.
- static void [ReplyAbortM](#) (struct libalias *la, struct in_addr ip_addr, uint16_t port, uint32_t vtag)
Send an abortM.
- static void [ReplyErrorM](#) (struct libalias *la, struct in_addr ip_addr, uint16_t port, uint32_t vtag)
Send an ErrorM.
- static u_int [StartPointLocal](#) (uint32_t l_vtag, uint16_t l_port)
Calculates the HASH value for lookups in the local look-up table.
- static u_int [StartPointGlobal](#) (uint32_t g_vtag, uint16_t g_port)
Calculates the HASH value for lookups in the global look-up table.
- static struct [sctp_nat_assoc](#) * [FindSctpLocal](#) (struct libalias *la, struct in_addr l_addr, uint32_t l_vtag, uint16_t l_port)
Find the SCTP association given the local address, port and vtag.
- static struct [sctp_nat_assoc](#) * [FindSctpGlobal](#) (struct libalias *la, uint32_t g_vtag, uint16_t g_port)
Find the SCTP association given the global port and vtag.
- static struct [sctp_nat_assoc](#) * [FindSctpLocalT](#) (struct libalias *la, uint32_t l_vtag, uint16_t g_port)
Find the SCTP association for a T-Flag message (given the global port and local vtag).
- static struct [sctp_nat_assoc](#) * [FindSctpGlobalT](#) (struct libalias *la, uint32_t g_vtag, uint16_t l_port)
Find the SCTP association for a T-Flag message (given the local port and global vtag).
- static int [AddSctpAssocLocal](#) (struct libalias *la, struct [sctp_nat_assoc](#) *assoc)
Add the sctp association information to the local look up table.
- static int [AddSctpAssocGlobal](#) (struct libalias *la, struct [sctp_nat_assoc](#) *assoc)
Add the sctp association information to the global look up table.
- static void [RmSctpAssoc](#) (struct libalias *la, struct [sctp_nat_assoc](#) *assoc)
Remove the sctp association information from the look up table.

- static void `sctp_AddTimeOut` (struct libalias *la, struct `sctp_nat_assoc` *assoc)
Add a timeout to the timer queue.
- static void `sctp_ResetTimeOut` (struct libalias *la, struct `sctp_nat_assoc` *assoc, int newexp)
Reset timer in timer queue.
- void `sctp_CheckExp` (struct libalias *la)
Check timer Q against current time.
- static void `sctp_RmTimeOut` (struct libalias *la, struct `sctp_nat_assoc` *assoc)
Remove association from timer queue.
- static __inline void * `sctp_NextChunkHdr` (struct `sctp_chunkhdr` *ch)
Get a pointer to the next chunk's header.
- static __inline void * `sctp_FirstChunkHdr` (struct `sctphdr` *sh)
Get a pointer to the first chunk's header.
- static void `logSctperror` (struct libalias *la, char *errmsg, uint32_t vtag, int error)
Log sctp nat errors.
- static void `logSctpparse` (struct libalias *la, int direction, struct `sctp_nat_msg` *sm)
Log what the parser parsed.
- static void `logSctpassoc` (struct libalias *la, struct `sctp_nat_assoc` *assoc, char *s)
Log an SCTP association's details.
- static void `logTimerQ` (struct libalias *la)
Output timer queue to log.
- static void `logSctpGlobal` (struct libalias *la)
Output Global table to log.
- static void `logSctpLocal` (struct libalias *la)
Output Local table to log.
- void `SctpShowAliasStats` (struct libalias *la)
- static void `SctpAliasLog` (FILE *stream, const char *format,...)
- void `AliasSctpInit` (struct libalias *la)
Initialises the SCTP NAT Implementation.
- void `AliasSctpTerm` (struct libalias *la)
Cleans-up the SCTP NAT Implementation prior to unloading.
- int `SctpAliasIn` (struct libalias *la, struct ip *pip)
Handles the incoming SCTP packets.
- int `SctpAliasOut` (struct libalias *la, struct ip *pip)
Handles the outgoing SCTP packets.

4.1.1 Detailed Description

Copyright (c) 2008, Centre for Advanced Internet Architectures Swinburne University of Technology, Melbourne, Australia (CRICOS number 00111D).

Alias_sctp forms part of the libalias kernel module to handle Network Address Translation (NAT) for the SCTP protocol.

This software was developed by David A. Hayes with leadership and advice from Jason But

The design is outlined in CAIA technical report number 080618A (D. Hayes and J. But, "Alias_sctp Version 0.1: SCTP NAT implementation in IPFW")

Development is part of the CAIA SONATA project, proposed by Jason But and Grenville Armitage: <http://caia.swin.edu.au/urp/sonata/>

Project made possible through funding from CISCO Systems

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The names of the authors, the "Centre for Advanced Internet Architectures" and "Swinburne University of Technology" may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [alias_sctp.c](#).

4.1.2 Define Documentation

4.1.2.1 #define sn_calloc(n, x) calloc(n,x)

Definition at line 146 of file alias_sctp.c.

Referenced by AliasSctpInit().

4.1.2.2 #define sn_free(x) free(x)

Definition at line 147 of file alias_sctp.c.

Referenced by AddSctpAssocGlobal(), AddSctpAssocLocal(), AliasSctpTerm(), sctp_CheckExp(), SctpAliasIn(), and SctpAliasOut().

4.1.2.3 #define sn_malloc(x) malloc(x)

Definition at line 145 of file alias_sctp.c.

Referenced by SCTP_PktParser().

4.1.3 Function Documentation

4.1.3.1 static int AddSctpAssocGlobal (struct libalias * *la*, struct sctp_nat_assoc * *assoc*) [static]

Add the sctp association information to the global look up table.

Searches the global look-up table for an existing association with the same details. If a match exists and is ONLY in the global look-up table then this is a repeated INIT packet, we need to remove this association from the look-up table and add the new association

The new association is added to the head of the list and state is updated

Parameters:

la Pointer to the relevant libalias instance

assoc pointer to sctp association

Returns:

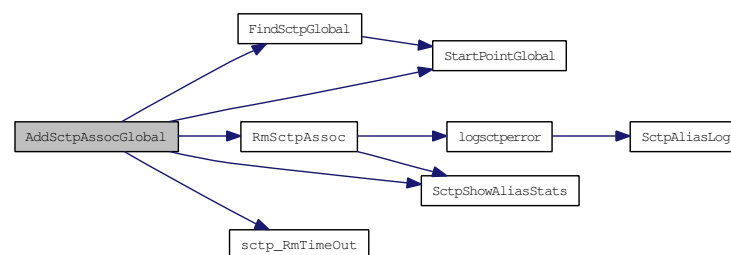
SN_ADD_OK | SN_ADD_CLASH

Definition at line 1131 of file alias_sctp.c.

References FindSctpGlobal(), sctp_nat_assoc::g_port, sctp_nat_assoc::g_vtag, sctp_nat_assoc::l_addr, sctp_nat_assoc::l_port, RmSctpAssoc(), sctp_RmTimeOut(), SctpShowAliasStats(), SN_ADD_CLASH, SN_ADD_OK, SN_BOTH_TBL, sn_free, SN_GLOBAL_TBL, StartPointGlobal(), and sctp_nat_assoc::TableRegister.

Referenced by ID_process(), and INi_process().

Here is the call graph for this function:



4.1.3.2 static int AddSctpAssocLocal (struct libalias * *la*, struct sctp_nat_assoc * *assoc*) [static]

Add the sctp association information to the local look up table.

Searches the local look-up table for an existing association with the same details. If a match exists and is ONLY in the local look-up table then this is a repeated INIT packet, we need to remove this association from the look-up table and add the new association

The new association is added to the head of the list and state is updated

Parameters:

la Pointer to the relevant libalias instance

assoc pointer to sctp association

Returns:

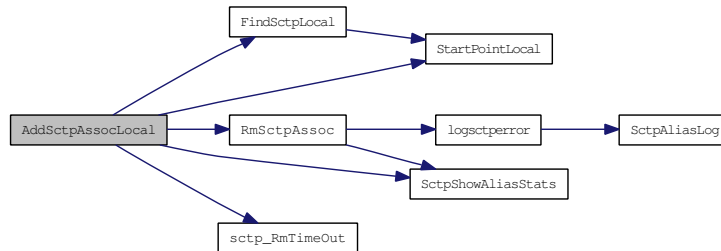
SN_ADD_OK | SN_ADD_CLASH

Definition at line 1080 of file alias_sctp.c.

References FindSctpLocal(), sctp_nat_assoc::g_port, sctp_nat_assoc::l_addr, sctp_nat_assoc::l_port, sctp_nat_assoc::l_vtag, RmSctpAssoc(), sctp_RmTimeOut(), SctpShowAliasStats(), SN_ADD_CLASH, SN_ADD_OK, SN_BOTH_TBL, sn_free, SN_LOCAL_TBL, StartPointLocal(), and sctp_nat_assoc::TableRegister.

Referenced by ID_process(), and INi_process().

Here is the call graph for this function:



4.1.3.3 void AliasSctpInit (struct libalias * la)

Initialises the SCTP NAT Implementation.

Creates the look-up tables and the time queue and initialises all state variables

Parameters:

la Pointer to the relevant libalias instance

Definition at line 176 of file alias_sctp.c.

References MAX_SCTP_TIMERQ, sn_calloc, SN_CURTIME, SN_TABLE_GLOBAL_SIZE, and SN_TABLE_LOCAL_SIZE.

4.1.3.4 void AliasSctpTerm (struct libalias * la)

Cleans-up the SCTP NAT Implementation prior to unloading.

Removes all entries from the timer queue, freeing associations as it goes. We then free memory allocated to the look-up tables and the time queue

NOTE: We do not need to traverse the look-up tables as each association will always have an entry in the timer queue, freeing this memory once will free all memory allocated to entries in the look-up tables

Parameters:

la Pointer to the relevant libalias instance

Definition at line 213 of file alias_sctp.c.

References MAX_SCTP_TIMERQ, and sn_free.

4.1.3.5 static int CL_process (struct libalias * *la*, int *direction*, struct sctp_nat_assoc * *assoc*, struct sctp_nat_msg * *sm*) [static]

Process sctp message while association is in the process of closing.

This function waits for a ShutComp to close the association. Depending on the the setting of SN_SHUTDOWN it may not remove the association immediately, but leave it up until SN_X_T. Only ShutComp, ShutAck, and Abort packets are permitted in this state. All other packets are dropped.

Parameters:

la Pointer to the relevant libalias instance

direction SN_TO_LOCAL | SN_TO_GLOBAL

assoc current sctp association

sm current sctp message

Returns:

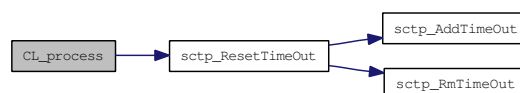
SN_NAT_PKT | SN_DROP_PKT

Definition at line 908 of file alias_sctp.c.

References sctp_nat_msg::msg, sctp_ResetTimeout(), sctpAbort, sctpShutAck, sctpShutComp, SN_C_T, SN_CL, SN_DROP_PKT, SN_NAT_PKT, SN_RM, SN_SHUTDOWN, SN_X_T, and sctp_nat_assoc::state.

Referenced by ProcessSctpMsg().

Here is the call graph for this function:



4.1.3.6 static struct sctp_nat_assoc * FindSctpGlobal (struct libalias * *la*, uint32_t *g_vtag*, uint16_t *g_port*) [static, read]

Find the SCTP association given the global port and vtag.

Searches the global look-up table for the association entry matching the provided global <port:vtag> tuple (address is not used as we match an association against all global addresses)

Parameters:

la Pointer to the relevant libalias instance

g_vtag global vtag

g_port global port

Returns:

pointer to association or NULL

Definition at line 983 of file alias_sctp.c.

References sctp_nat_assoc::g_port, sctp_nat_assoc::g_vtag, and StartPointGlobal().

Referenced by AddSctpAssocGlobal(), and SCTP_PktParser().

Here is the call graph for this function:



4.1.3.7 static struct sctp_nat_assoc * FindSctpGlobalT (struct libalias * *la*, uint32_t *g_vtag*, uint16_t *l_port*) [static, read]

Find the SCTP association for a T-Flag message (given the local port and global vtag).

Searches the global look-up table for a unique association entry matching the provided local port and global vtag information

Parameters:

la Pointer to the relevant libalias instance

g_vtag global vtag

g_port global port

Returns:

pointer to association or NULL

Definition at line 1048 of file alias_sctp.c.

References sctp_nat_assoc::l_port, sctp_nat_assoc::l_vtag, and StartPointLocal().

Referenced by SCTP_PktParser().

Here is the call graph for this function:



4.1.3.8 static struct sctp_nat_assoc * FindSctpLocal (struct libalias * *la*, struct in_addr *l_addr*, uint32_t *l_vtag*, uint16_t *l_port*) [static, read]

Find the SCTP association given the local address, port and vtag.

Searches the local look-up table for the association entry matching the provided local <address:port:vtag> tuple

Parameters:

la Pointer to the relevant libalias instance
l_addr local address
l_vtag local Vtag
l_port local Port

Returns:

pointer to association or NULL

Definition at line 952 of file alias_sctp.c.

References sctp_nat_assoc::l_addr, sctp_nat_assoc::l_port, sctp_nat_assoc::l_vtag, and StartPointLocal().

Referenced by AddSctpAssocLocal(), and SCTP_PktParser().

Here is the call graph for this function:



4.1.3.9 static struct sctp_nat_assoc * FindSctpLocalT (struct libalias * la, uint32_t l_vtag, uint16_t g_port) [static, read]

Find the SCTP association for a T-Flag message (given the global port and local vtag).

Searches the local look-up table for a unique association entry matching the provided global port and local vtag information

Parameters:

la Pointer to the relevant libalias instance
l_addr local address
l_vtag local Vtag
l_port local Port

Returns:

pointer to association or NULL

Definition at line 1013 of file alias_sctp.c.

References sctp_nat_assoc::g_port, sctp_nat_assoc::g_vtag, and StartPointGlobal().

Referenced by SCTP_PktParser().

Here is the call graph for this function:



4.1.3.10 `static int ID_process (struct libalias * la, int direction, struct sctp_nat_assoc * assoc, struct sctp_nat_msg * sm) [static]`

Process sctp message while in the Idle state.

This function looks for an Incoming Init or AddIp message All other sctp messages are invalid when in SN_ID, and are dropped.

Parameters:

la Pointer to the relevant libalias instance
direction SN_TO_LOCAL | SN_TO_GLOBAL
assoc current sctp association
sm current sctp message

Returns:

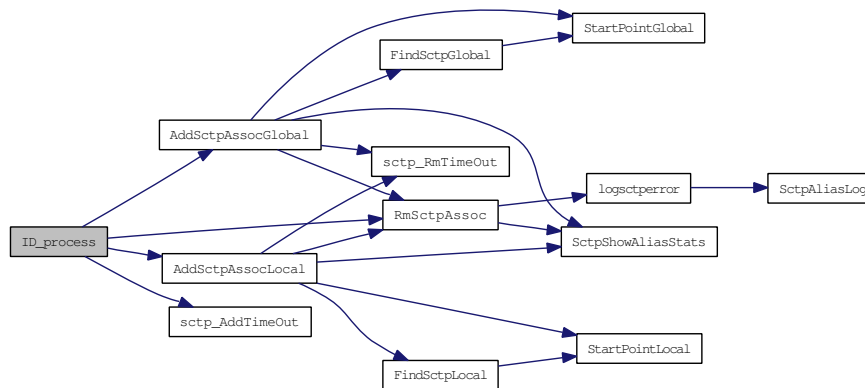
SN_NAT_PKT | SN_DROP_PKT | SN_REPLY_ABORT | SN_REPLY_ERROR

Definition at line 704 of file alias_sctp.c.

References AddSctpAssocGlobal(), AddSctpAssocLocal(), sctp_nat_assoc::exp, sctp_nat_assoc::g_port, sctp_nat_assoc::g_vtag, sctpChunkOfInt::Init, sctp_nat_msg::ip_hdr, sctp_nat_assoc::l_addr, sctp_nat_assoc::l_port, sctp_nat_assoc::l_vtag, sctp_nat_msg::msg, RmSctpAssoc(), sctp_AddTimeOut(), sctp_nat_msg::sctp_hdr, sctpAddIp, sctp_nat_msg::sctpchnk, sctpInit, SN_DROP_PKT, SN_I_T, SN_INa, SN_INi, SN_NAT_PKT, SN_REPLY_ABORT, SN_REPLY_ERROR, SN_TO_GLOBAL, SN_TO_LOCAL, and sctp_nat_assoc::state.

Referenced by ProcessSctpMsg().

Here is the call graph for this function:



4.1.3.11 `static int INa_process (struct libalias * la, int direction, struct sctp_nat_assoc * assoc, struct sctp_nat_msg * sm) [static]`

Process sctp message while waiting for an AddIpAck message.

Only an AddIPack, resent AddIP, or an Abort message are valid all other sctp packets are dropped

Parameters:

la Pointer to the relevant libalias instance

direction SN_TO_LOCAL | SN_TO_GLOBAL

assoc current sctp association

sm current sctp message

Returns:

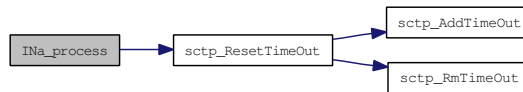
SN_NAT_PKT | SN_DROP_PKT

Definition at line 835 of file alias_sctp.c.

References sctp_nat_msg::msg, sctp_ResetTimeout(), sctpAbort, sctpAddIp, sctpAddIpAck, SN_DROP_PKT, SN_I_T, SN_NAT_PKT, SN_RM, SN_U_T, SN_UP, and sctp_nat_assoc::state.

Referenced by ProcessSctpMsg().

Here is the call graph for this function:



4.1.3.12 static int INi_process (struct libalias * la, int direction, struct sctp_nat_assoc * assoc, struct sctp_nat_msg * sm) [static]

Process sctp message while waiting for an InitAck message.

Only an InitAck, resent Init, or an Abort sctp packet are valid in this state. All other packets are dropped.

Parameters:

la Pointer to the relevant libalias instance

direction SN_TO_LOCAL | SN_TO_GLOBAL

assoc current sctp association

sm current sctp message

Returns:

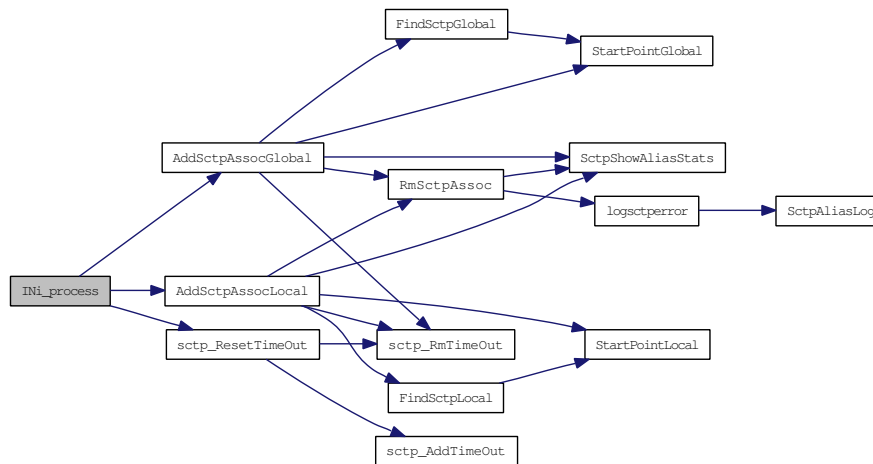
SN_NAT_PKT | SN_DROP_PKT | SN_REPLY_ABORT

Definition at line 781 of file alias_sctp.c.

References AddSctpAssocGlobal(), AddSctpAssocLocal(), sctp_nat_assoc::g_vtag, sctpChunkOfInt::Init, sctp_nat_msg::ip_hdr, sctp_nat_assoc::l_addr, sctp_nat_assoc::l_vtag, sctp_nat_msg::msg, sctp_ResetTimeout(), sctpAbort, sctp_nat_msg::sctpchnk, sctpInit, sctpInitAck, SN_DROP_PKT, SN_I_T, SN_NAT_PKT, SN_RM, SN_SEND_ABORT, SN_TO_GLOBAL, SN_TO_LOCAL, SN_U_T, SN_UP, and sctp_nat_assoc::state.

Referenced by ProcessSctpMsg().

Here is the call graph for this function:



4.1.3.13 static void logscptpassoc (struct libalias * *la*, struct sctp_nat_assoc * *assoc*, char * *s*) [static]

Log an SCTP association's details.

Parameters:

la Pointer to the relevant libalias instance

assoc pointer to sctp association

s Character that indicates the state of processing for this packet

Definition at line 1468 of file alias_sctp.c.

References sctp_nat_assoc::exp, sctp_nat_assoc::g_port, sctp_nat_assoc::g_vtag, sctp_nat_assoc::l_addr, sctp_nat_assoc::l_port, sctp_nat_assoc::l_vtag, SctpAliasLog(), SN_CL, SN_ID, SN_INa, SN_INi, SN_RM, SN_UP, sctp_nat_assoc::state, and sctp_nat_assoc::TableRegister.

Referenced by logSctpGlobal(), logSctpLocal(), logTimerQ(), SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.14 static void logscpterror (struct libalias * *la*, char * *errmsg*, uint32_t *vtag*, int *error*) [static]

Log sctp nat errors.

Code for debugging.

This code provides detailed log messages

Parameters:

- la* Pointer to the relevant libalias instance
- errmsg* Error message to be logged
- vtag* Current Vtag
- error* Error number

Definition at line 1396 of file alias_sctp.c.

References SctpAliasLog().

Referenced by RmSctpAssoc(), sctp_CheckExp(), SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:

**4.1.3.15 static void logSctpGlobal (struct libalias * la) [static]**

Output Global table to log.

Parameters:

- la* Pointer to the relevant libalias instance

Definition at line 1509 of file alias_sctp.c.

References logSctpAssoc(), SctpAliasLog(), and SN_TABLE_GLOBAL_SIZE.

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:

**4.1.3.16 static void logSctpLocal (struct libalias * la) [static]**

Output Local table to log.

Parameters:

- la* Pointer to the relevant libalias instance

Definition at line 1527 of file alias_sctp.c.

References logSctpAssoc(), SctpAliasLog(), and SN_TABLE_LOCAL_SIZE.

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.17 static void logsetpparse (struct libalias * *la*, int *direction*, struct sctp_nat_msg * *sm*) [static]

Log what the parser parsed.

Parameters:

la Pointer to the relevant libalias instance

direction Direction of packet

sm pointer to the [sctp_nat_msg](#) structure

Definition at line 1412 of file alias_sctp.c.

References sctp_nat_msg::msg, sctpAbort, sctpAddIp, sctpAddIpAck, SctpAliasLog(), sctpInit, sctpInitAck, sctpOther, sctpShutAck, sctpShutComp, SN_TO_GLOBAL, and SN_TO_LOCAL.

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.18 static void logTimerQ (struct libalias * *la*) [static]

Output timer queue to log.

Parameters:

la Pointer to the relevant libalias instance

Definition at line 1545 of file alias_sctp.c.

References logSctpAssoc(), MAX_SCTP_TIMERQ, and SctpAliasLog().

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.19 static int ProcessSctpMsg (struct libalias * *la*, int *direction*, struct sctp_nat_msg * *sm*, struct sctp_nat_assoc * *assoc*) [static]

Process SCTP message.

This function is the base state machine. It calls the processing engine for each state.

Parameters:

la Pointer to the relevant libalias instance
direction SN_TO_LOCAL or SN_TO_GLOBAL
assoc current sctp association
sm current sctp message

Returns:

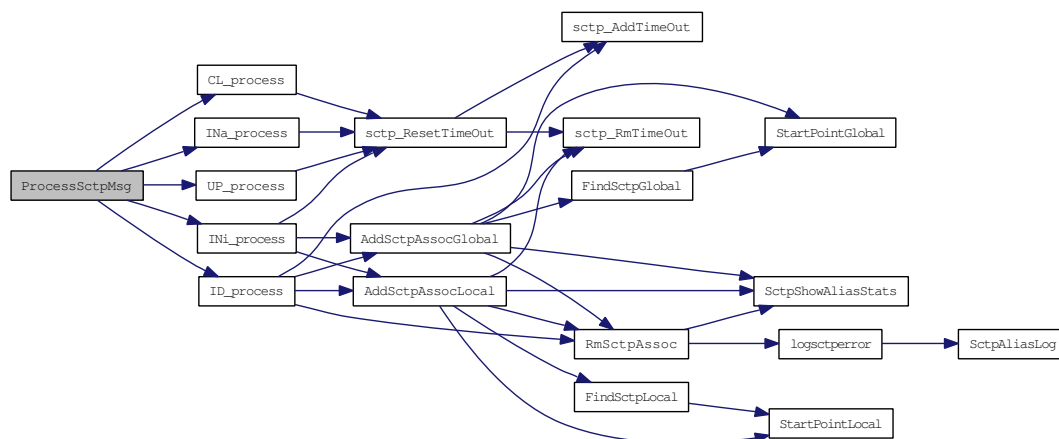
SN_DROP_PKT | SN_NAT_PKT | SN_REPLY_ABORT | SN_REPLY_ERROR | SN_PROCESSING_ERROR

Definition at line 605 of file alias_sctp.c.

References CL_process(), ID_process(), INa_process(), INi_process(), SN_CL, SN_ID, SN_INa, SN_INi, SN_NAT_PKT, SN_PROCESSING_ERROR, SN_RM, SN_UP, sctp_nat_assoc::state, and UP_process().

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.20 static void ReplyAbortM (struct libalias * *la*, struct in_addr *ip_addr*, uint16_t *port*, uint32_t *vtag*) [static]

Send an abortM.

Parameters:

la Pointer to the relevant libalias instance
ip_addr address to send the abort to
port port to send the abort to

vtag vtag to send the abort to

Definition at line 642 of file alias_sctp.c.

References SctpAliasLog().

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.21 static void ReplyErrorM (struct libalias * *la*, struct in_addr *ip_addr*, uint16_t *port*, uint32_t *vtag*) [static]

Send an ErrorM.

Parameters:

la Pointer to the relevant libalias instance

ip_addr address to send the error to

port port to send the error to

vtag vtag to send the error to

Definition at line 671 of file alias_sctp.c.

References SctpAliasLog().

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.22 static void RmSctpAssoc (struct libalias * *la*, struct sctp_nat_assoc * *assoc*) [static]

Remove the sctp association information from the look up table.

For each of the two (local/global) look-up tables, remove the association from that table IF it has been registered in that table.

NOTE: The calling code is responsible for freeing memory allocated to the association structure itself

NOTE: The association is NOT removed from the timer queue

Parameters:

la Pointer to the relevant libalias instance

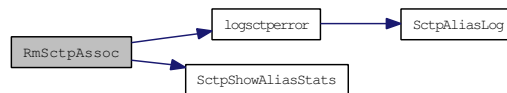
assoc pointer to sctp association

Definition at line 1173 of file alias_sctp.c.

References `logscpterror()`, `SctpShowAliasStats()`, `SN_GLOBAL_TBL`, `SN_LOCAL_TBL`, and `sctp_nat_assoc::TableRegister`.

Referenced by `AddSctpAssocGlobal()`, `AddSctpAssocLocal()`, `ID_process()`, `sctp_CheckExp()`, `SctpAliasIn()`, and `SctpAliasOut()`.

Here is the call graph for this function:



4.1.3.23 `static void sctp_AddTimeOut (struct libalias * la, struct sctp_nat_assoc * assoc)` [static]

Add a timeout to the timer queue.

Timer Queue management functions

These are designed to operate efficiently with a minimum of interaction with the queues

Once a timeout is set in the queue it will not be altered in the queue unless it has to be changed to a shorter time (usually only for aborts and closing). On a queue timeout, the real expiry time is checked, and if not less than the timeout it is requeued (O(1)) at its later time. This is especially important for normal packets sent during an association. When a timer expires, it is updated to its new expiration time if necessary, or processed as a timeout. This means that while in UP state, the timing queue is only altered every U_T (every few minutes) for a particular association.

Parameters:

la

assoc

Definition at line 1287 of file alias_sctp.c.

References `sctp_nat_assoc::exp`, `sctp_nat_assoc::exp_loc`, and `MAX_SCTP_TIMERQ`.

Referenced by `ID_process()`, `sctp_CheckExp()`, and `sctp_ResetTimeOut()`.

4.1.3.24 `void sctp_CheckExp (struct libalias * la)`

Check timer Q against current time.

Parameters:

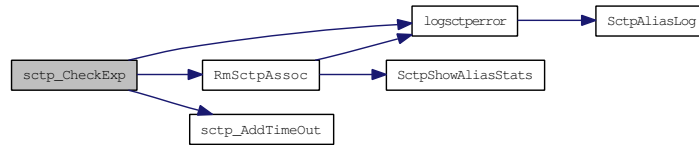
la Pointer to the relevant libalias instance

Definition at line 1345 of file alias_sctp.c.

References `_SCTP_NAT_DEBUG`, `sctp_nat_assoc::exp`, `sctp_nat_assoc::g_vtag`, `logscpterror()`, `MAX_SCTP_TIMERQ`, `RmSctpAssoc()`, `sctp_AddTimeOut()`, `SN_CL`, `SN_CURTIME`, `sn_free`, and `sctp_nat_assoc::state`.

Referenced by `SctpAliasIn()`, and `SctpAliasOut()`.

Here is the call graph for this function:



4.1.3.25 `static __inline void * sctp_FirstChunkHdr (struct sctphdr * sh) [static]`

Get a pointer to the first chunk's header.

Parameters:

sh - pointer to the sctp common header

Returns:

pointer to next chunk's header

Definition at line 1258 of file alias_sctp.c.

Referenced by Sctp_PktParser().

4.1.3.26 `static __inline void * sctp_NextChunkHdr (struct sctp_chunkhdr * ch) [static]`

Get a pointer to the next chunk's header.

Parameters:

ch - pointer to the current chunk's header

Returns:

pointer to next chunk's header

Definition at line 1244 of file alias_sctp.c.

Referenced by Sctp_PktParser().

4.1.3.27 `static int Sctp_PktParser (struct libalias * la, int direction, struct ip * pip, struct sctp_nat_msg * sm, struct sctp_nat_assoc ** passoc) [static]`

parses Sctp packets for the key Sctp chunk that will be processed

This module parses Sctp packets for the key Sctp chunk that will be processed The module completes the [sctp_nat_msg](#) structure and either retrieves the relevant [sctp_nat_assoc](#) or creates a new [sctp_nat_assoc](#) at state SN_ID

Parameters:

la Pointer to the relevant libalias instance

ip

ah

Returns:

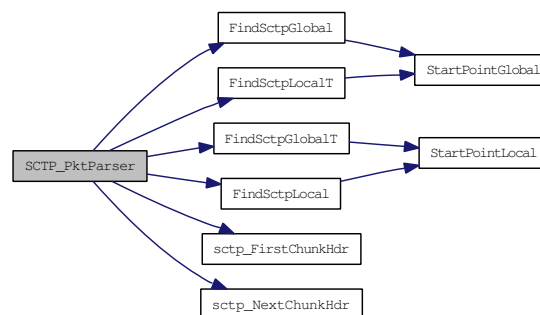
SN_PARSE_OK | SN_PARSE_ERROR_*

Definition at line 438 of file alias_sctp.c.

References sctpChunkOfInt::Asconf, FindSctpGlobal(), FindSctpGlobalT(), FindSctpLocal(), FindSctpLocalT(), sctpChunkOfInt::Init, sctpChunkOfInt::InitAck, sctp_nat_msg::ip_hdr, sctp_nat_msg::msg, sctp_FirstChunkHdr(), sctp_nat_msg::sctp_hdr, sctp_NextChunkHdr(), sctpAbort, sctpAddIp, sctpAddIpAck, sctp_nat_msg::sctpchnk, sctpInit, sctpInitAck, sctpOther, sctpShutAck, sctpShutComp, SN_ID, sn_malloc, SN_NULL_TBL, SN_PARSE_ERROR_AS_MALLOC, SN_PARSE_ERROR_CHHL, SN_PARSE_ERROR_CHUNK, SN_PARSE_ERROR_DIR, SN_PARSE_ERROR_IPSHL, SN_PARSE_ERROR_VTAG, SN_PARSE_OK, SN_TO_GLOBAL, and SN_TO_LOCAL.

Referenced by SctpAliasIn(), and SctpAliasOut().

Here is the call graph for this function:



4.1.3.28 static void sctp_ResetTimeout (struct libalias * *la*, struct sctp_nat_assoc * *assoc*, int *newexp*) [static]

Reset timer in timer queue.

Reset a timer in the timer queue only if the timeout will be earlier

Parameters:

la Pointer to the relevant libalias instance

assoc pointer to sctp association

newexp New expiration time

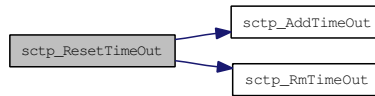
Only change timer Q if the timeout will be earlier

Definition at line 1323 of file alias_sctp.c.

References sctp_nat_assoc::exp, sctp_AddTimeOut(), and sctp_RmTimeOut().

Referenced by CL_process(), INa_process(), INi_process(), and UP_process().

Here is the call graph for this function:



4.1.3.29 `static void sctp_RmTimeout (struct libalias * la, struct sctp_nat_assoc * assoc)` [static]

Remove association from timer queue.

Parameters:

la Pointer to the relevant libalias instance

assoc pointer to sctp association

Definition at line 1305 of file `alias_sctp.c`.

Referenced by `AddSctpAssocGlobal()`, `AddSctpAssocLocal()`, `sctp_ResetTimeout()`, `SctpAliasIn()`, and `SctpAliasOut()`.

4.1.3.30 `int SctpAliasIn (struct libalias * la, struct ip * pip)`

Handles the incoming SCTP packets.

Parameters:

la Pointer to the relevant libalias instance

ip Pointer to IP packet to process

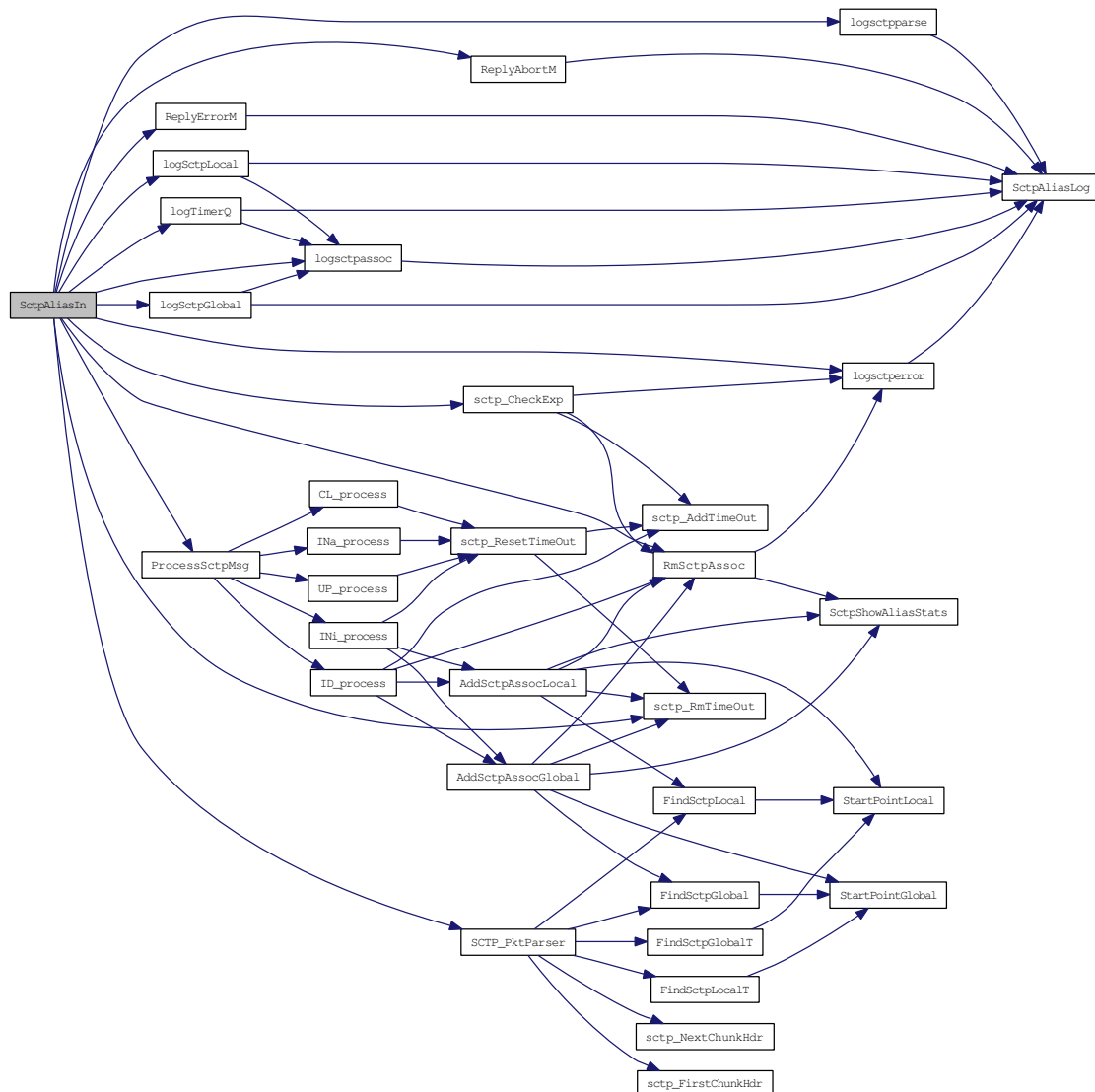
Returns:

`PKT_ALIAS_OK` | `PKT_ALIAS_IGNORE` | `PKT_ALIAS_ERROR`

Definition at line 242 of file `alias_sctp.c`.

References `_SCTP_NAT_DEBUG`, `sctpChunkOfInt::Init`, `sctp_nat_msg::ip_hdr`, `sctp_nat_assoc::l_addr`, `logSctpAssoc()`, `logSctpError()`, `logSctpGlobal()`, `logSctpLocal()`, `logSctpParse()`, `logTimerQ()`, `ProcessSctpMsg()`, `ReplyAbortM()`, `ReplyErrorM()`, `RmSctpAssoc()`, `sctp_CheckExp()`, `sctp_nat_msg::sctp_hdr`, `SCTP_PktParser()`, `sctp_RmTimeout()`, `sctp_nat_msg::sctpchnk`, `SN_DROP_PKT`, `sn_free`, `SN_NAT_PKT`, `SN_PARSE_OK`, `SN_REPLY_ABORT`, `SN_REPLY_ERROR`, `SN_RM`, `SN_SEND_ABORT`, `SN_TO_LOCAL`, `sctp_nat_assoc::state`, and `sctp_nat_assoc::TableRegister`.

Here is the call graph for this function:



4.1.3.31 static void SctpAliasLog (FILE * *stream*, const char * *format*, ...) [static]

Definition at line 1583 of file alias_sctp.c.

Referenced by logSctpPassoc(), logSctpError(), logSctpGlobal(), logSctpLocal(), logSctpParse(), logTimerQ(), ReplyAbortM(), and ReplyErrorM().

4.1.3.32 int SctpAliasOut (struct libalias * *la*, struct ip * *pip*)

Handles the outgoing SCTP packets.

Parameters:

la Pointer to the relevant libalias instance

ip Pointer to IP packet to process

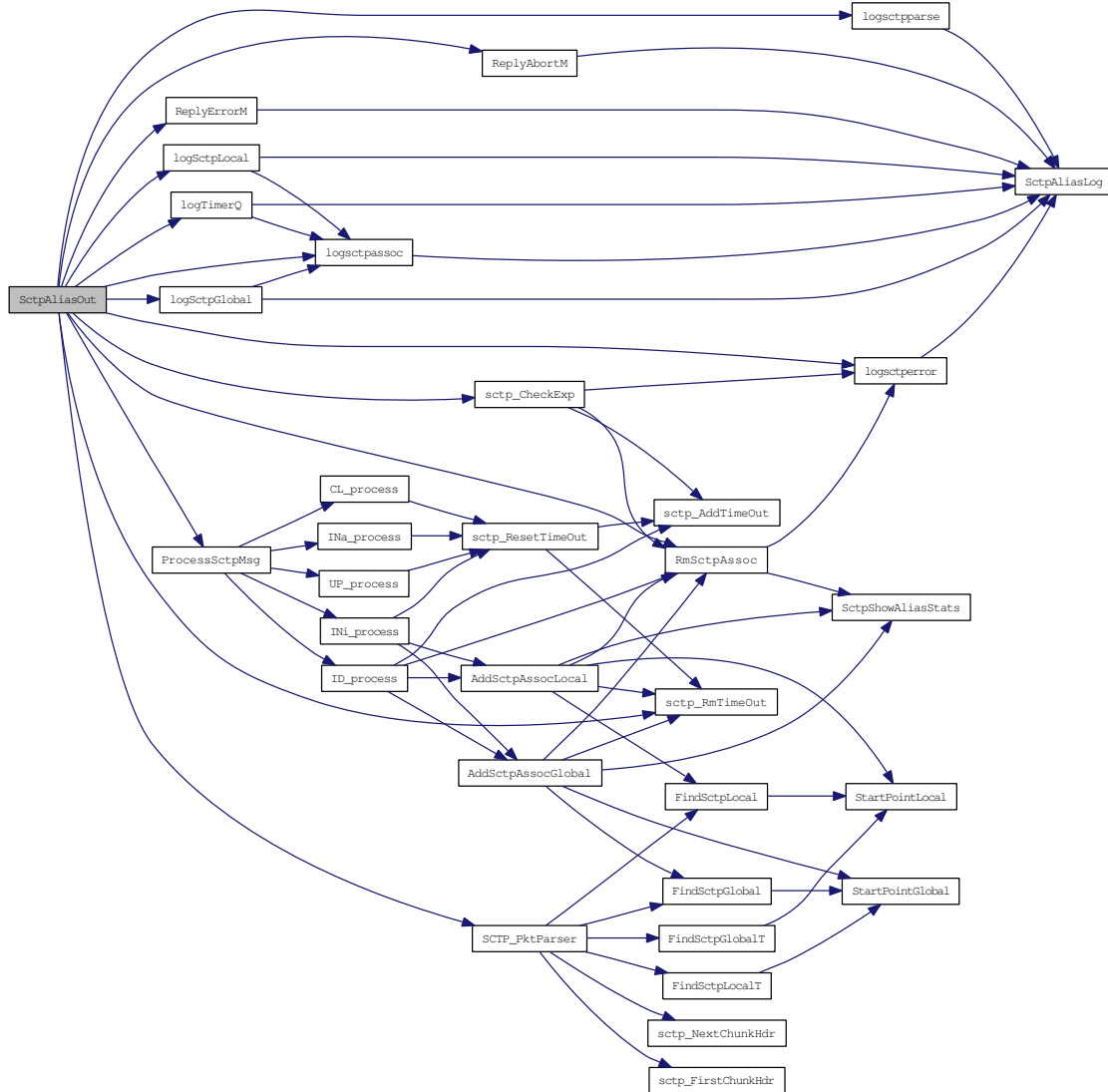
Returns:

PKT_ALIAS_OK | PKT_ALIAS_IGNORE | PKT_ALIAS_ERROR

Definition at line 337 of file alias_sctp.c.

References `_SCTP_NAT_DEBUG`, `sctpChunkOfInt::Init`, `sctp_nat_msg::ip_hdr`, `logsctpassoc()`, `logsctperror()`, `logSctpGlobal()`, `logSctpLocal()`, `logsctpparse()`, `logTimerQ()`, `ProcessSctpMsg()`, `ReplyAbortM()`, `ReplyErrorM()`, `RmSctpAssoc()`, `sctp_CheckExp()`, `sctp_nat_msg::sctp_hdr`, `SCTP_PktParser()`, `sctp_RmTimeOut()`, `sctp_nat_msg::sctpchnk`, `SN_DROP_PKT`, `sn_free`, `SN_NAT_PKT`, `SN_PARSE_OK`, `SN_REPLY_ABORT`, `SN_REPLY_ERROR`, `SN_RM`, `SN_SEND_ABORT`, `SN_TO_GLOBAL`, `sctp_nat_assoc::state`, and `sctp_nat_assoc::TableRegister`.

Here is the call graph for this function:



4.1.3.33 void SctpShowAliasStats (struct libalias * *la*)

Referenced by AddSctpAssocGlobal(), AddSctpAssocLocal(), and RmSctpAssoc().

4.1.3.34 static u_int StartPointGlobal (uint32_t *g_vtag*, uint16_t *g_port*) [static]

Calculates the HASH value for lookups in the global look-up table.

Parameters:

g_vtag global vtag

g_port global port

Returns:

starting point in table

Definition at line 1229 of file alias_sctp.c.

References SN_TABLE_GLOBAL_SIZE.

Referenced by AddSctpAssocGlobal(), FindSctpGlobal(), and FindSctpLocalT().

4.1.3.35 static u_int StartPointLocal (uint32_t *l_vtag*, uint16_t *l_port*) [static]

Calculates the HASH value for lookups in the local look-up table.

Parameters:

l_vtag local vtag

l_port local port

Returns:

starting point in table

Definition at line 1212 of file alias_sctp.c.

References SN_TABLE_LOCAL_SIZE.

Referenced by AddSctpAssocLocal(), FindSctpGlobalT(), and FindSctpLocal().

4.1.3.36 static int UP_process (struct libalias * *la*, int *direction*, struct sctp_nat_assoc * *assoc*, struct sctp_nat_msg * *sm*) [static]

Process sctp messages while association is UP redirecting packets.

While in the SN_UP state, all packets for the particular association are passed. Only a ShutAck or an Abort will cause a change of state.

Parameters:

la Pointer to the relevant libalias instance

direction SN_TO_LOCAL | SN_TO_GLOBAL

assoc current sctp association

sm current sctp message

Returns:

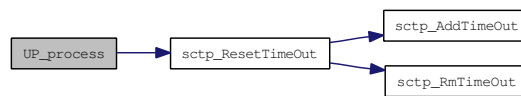
SN_NAT_PKT | SN_DROP_PKT

Definition at line 871 of file alias_sctp.c.

References sctp_nat_msg::msg, sctp_ResetTimeOut(), sctpAbort, sctpShutAck, SN_C_T, SN_CL, SN_DROP_PKT, SN_NAT_PKT, SN_RM, SN_U_T, and sctp_nat_assoc::state.

Referenced by ProcessSctpMsg().

Here is the call graph for this function:



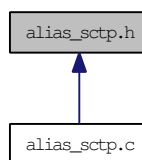
4.2 alias_sctp.h File Reference

```
#include <sys/param.h>
#include <sys/types.h>
#include <sys/queue.h>
#include <sys/time.h>
#include <netinet/in_system.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <machine/cpufunc.h>
#include <machine/cpu.h>
#include <netinet/sctp.h>
#include <netinet/sctp_header.h>
#include <stdlib.h>
#include <stdio.h>
#include <curses.h>
```

Include dependency graph for alias_sctp.h:



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



Data Structures

- struct [sctp_nat_assoc](#)
sctp association information
- union [sctpChunkOfInt](#)
SCTP chunk of interest.
- struct [sctp_nat_msg](#)
SCTP message.
- struct [sctp_nat_timer](#)
sctp nat timer queue structure

Defines

- #define [SCTP_PACKED](#) __attribute__((packed))
- #define [SCTP_UNUSED](#) __attribute__((unused))
- #define [LINK_SCTP](#) IPPROTO_SCTP
- #define [SN_SHUTDOWN](#) 0
- #define [SN_TABLE_LOCAL_SIZE](#) 1001
- #define [SN_TABLE_GLOBAL_SIZE](#) 1001
- #define [_SCTP_NAT_DEBUG](#) 0
- #define [SN_TO_LOCAL](#) 0
- #define [SN_TO_GLOBAL](#) 1
- #define [SN_GLOBAL_TBL](#) 0x02
- #define [SN_LOCAL_TBL](#) 0x01
- #define [SN_BOTH_TBL](#) 0x03
- #define [SN_NULL_TBL](#) 0x00
- #define [sctpInit](#) 0x0001
- #define [sctpInitAck](#) 0x0002
- #define [sctpAbort](#) 0x0000
- #define [sctpShutAck](#) 0x0010
- #define [sctpShutComp](#) 0x0020
- #define [sctpAddIp](#) 0x0100
- #define [sctpAddIpAck](#) 0x0200
- #define [sctpOther](#) 0xFFFF
- #define [SN_NAT_PKT](#) 0
- #define [SN_DROP_PKT](#) 1
- #define [SN_REPLY_ABORT](#) 2
- #define [SN_REPLY_ERROR](#) 3
- #define [SN_SEND_ABORT](#) 4
- #define [SN_ADD_OK](#) 0
- #define [SN_ADD_CLASH](#) 1
- #define [SN_PROCESSING_ERROR](#) 5
- #define [SN_PARSE_OK](#) 0
- #define [SN_PARSE_ERROR_IPSHL](#) 1
- #define [SN_PARSE_ERROR_AS_MALLOC](#) 2
- #define [SN_PARSE_ERROR_CHHL](#) 3
- #define [SN_PARSE_ERROR_DIR](#) 4
- #define [SN_PARSE_ERROR_VTAG](#) 5
- #define [SN_PARSE_ERROR_CHUNK](#) 6
- #define [SN_ID](#) 0x0000
- #define [SN_INi](#) 0x0010
- #define [SN_INa](#) 0x0020
- #define [SN_UP](#) 0x0100
- #define [SN_CL](#) 0x1000
- #define [SN_RM](#) 0x2000
- #define [SN_UT](#) 600
- #define [SN_CURTIME](#) la → timeStamp
- #define [SN_I_T](#) SN_CURTIME+60
- #define [SN_U_T](#) SN_CURTIME+SN_UT
- #define [SN_C_T](#) SN_CURTIME+60
- #define [SN_X_T](#) SN_CURTIME+3
- #define [SN_A_T](#) SN_CURTIME+1
- #define [MAX_SCTP_TIMERQ](#) SN_UT+10

4.2.1 Detailed Description

Copyright (c) 2008, Centre for Advanced Internet Architectures Swinburne University of Technology, Melbourne, Australia (CRICOS number 00111D).

Alias_sctp forms part of the libalias kernel module to handle Network Address Translation (NAT) for the SCTP protocol.

This software was developed by David A. Hayes with leadership and advice from Jason But

The design is outlined in CAIA technical report number 080618A (D. Hayes and J. But, "Alias_sctp Version 0.1: SCTP NAT implementation in IPFW")

Development is part of the CAIA SONATA project, proposed by Jason But and Grenville Armitage: <http://caia.swin.edu.au/urp/sonata/>

Project made possible through funding from CISCO Systems

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The names of the authors, the "Centre for Advanced Internet Architectures" and "Swinburne University of Technology" may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHORS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Definition in file [alias_sctp.h](#).

4.2.2 Define Documentation

4.2.2.1 #define _SCTP_NAT_DEBUG 0

Definition at line 128 of file [alias_sctp.h](#).

Referenced by [sctp_CheckExp\(\)](#), [SctpAliasIn\(\)](#), and [SctpAliasOut\(\)](#).

4.2.2.2 #define LINK_SCTP IPPROTO_SCTP

Definition at line 114 of file [alias_sctp.h](#).

4.2.2.3 #define MAX_SCTP_TIMERQ SN_UT+10

max timer queue size is max time + 10s

Definition at line 251 of file alias_sctp.h.

Referenced by AliasSctpInit(), AliasSctpTerm(), logTimerQ(), sctp_AddTimeOut(), and sctp_CheckExp().

4.2.2.4 #define SCTP_PACKED __attribute__((packed))

These are defined in sctp_os_bsd.h, but it can't be included due to its local file inclusion, so I'm defining them here.

Definition at line 96 of file alias_sctp.h.

4.2.2.5 #define SCTP_UNUSED __attribute__((unused))

Definition at line 99 of file alias_sctp.h.

4.2.2.6 #define sctpAbort 0x0000

a packet containing an ABORT chunk

Definition at line 143 of file alias_sctp.h.

Referenced by CL_process(), INa_process(), INi_process(), logsctpparse(), SCTP_PktParser(), and UP_process().

4.2.2.7 #define sctpAddIp 0x0100

a packet containing an ASCONF chunk with ADDIP

Definition at line 146 of file alias_sctp.h.

Referenced by ID_process(), INa_process(), logsctpparse(), and SCTP_PktParser().

4.2.2.8 #define sctpAddIpAck 0x0200

a packet containing an ASCONF chunk with a ADDIP-ACK

Definition at line 147 of file alias_sctp.h.

Referenced by INa_process(), logsctpparse(), and SCTP_PktParser().

4.2.2.9 #define sctpInit 0x0001

alias_sctp performs its processing based on a number of key messages a packet containing an INIT chunk

Definition at line 141 of file alias_sctp.h.

Referenced by ID_process(), INi_process(), logsctpparse(), and SCTP_PktParser().

4.2.2.10 #define sctpInitAck 0x0002

a packet containing an INIT-ACK chunk

Definition at line 142 of file alias_sctp.h.

Referenced by INi_process(), logsctpparse(), and SCTP_PktParser().

4.2.2.11 #define sctpOther 0xFFFF

a packet containing a chunk that is not of interest

Definition at line 148 of file alias_sctp.h.

Referenced by logsctpparse(), and SCTP_PktParser().

4.2.2.12 #define sctpShutAck 0x0010

a packet containing a SHUTDOWN-ACK chunk

Definition at line 144 of file alias_sctp.h.

Referenced by CL_process(), logsctpparse(), SCTP_PktParser(), and UP_process().

4.2.2.13 #define sctpShutComp 0x0020

a packet containing a SHUTDOWN-COMPLETE chunk

Definition at line 145 of file alias_sctp.h.

Referenced by CL_process(), logsctpparse(), and SCTP_PktParser().

4.2.2.14 #define SN_A_T SN_CURTIME+1

short abort timer to allow NAT before clearing DB

Definition at line 192 of file alias_sctp.h.

4.2.2.15 #define SN_ADD_CLASH 1

Clash when trying to add the assoc. info to the table

Definition at line 157 of file alias_sctp.h.

Referenced by AddSctpAssocGlobal(), and AddSctpAssocLocal().

4.2.2.16 #define SN_ADD_OK 0

Association added to the table

Definition at line 156 of file alias_sctp.h.

Referenced by AddSctpAssocGlobal(), and AddSctpAssocLocal().

4.2.2.17 #define SN_BOTH_TBL 0x03

assoc in both tables

Definition at line 135 of file alias_sctp.h.

Referenced by AddSctpAssocGlobal(), and AddSctpAssocLocal().

4.2.2.18 #define SN_C_T SN_CURTIME+60

state expiration time in seconds

Definition at line 186 of file alias_sctp.h.

Referenced by CL_process(), and UP_process().

4.2.2.19 #define SN_CL 0x1000

Closing state

Definition at line 179 of file alias_sctp.h.

Referenced by CL_process(), logsctpassoc(), ProcessSctpMsg(), sctp_CheckExp(), and UP_process().

4.2.2.20 #define SN_CURTIME la → timeStamp

current time based on the libalias timeStamp

Definition at line 183 of file alias_sctp.h.

Referenced by AliasSctpInit(), and sctp_CheckExp().

4.2.2.21 #define SN_DROP_PKT 1

drop packet (don't forward it)

Definition at line 151 of file alias_sctp.h.

Referenced by CL_process(), ID_process(), INa_process(), INi_process(), SctpAliasIn(), SctpAliasOut(), and UP_process().

4.2.2.22 #define SN_GLOBAL_TBL 0x02

assoc in global table

Definition at line 133 of file alias_sctp.h.

Referenced by AddSctpAssocGlobal(), and RmSctpAssoc().

4.2.2.23 #define SN_I_T SN_CURTIME+60

state expiration time in seconds

Definition at line 184 of file alias_sctp.h.

Referenced by ID_process(), INa_process(), and INi_process().

4.2.2.24 #define SN_ID 0x0000

Idle state

Definition at line 175 of file alias_sctp.h.

Referenced by logsctpassoc(), ProcessSctpMsg(), and SCTP_PktParser().

4.2.2.25 #define SN_INa 0x0020

Initialising, waiting for AddIpAck state

Definition at line 177 of file alias_sctp.h.

Referenced by ID_process(), logsctpassoc(), and ProcessSctpMsg().

4.2.2.26 #define SN_INi 0x0010

Initialising, waiting for InitAck state

Definition at line 176 of file alias_sctp.h.

Referenced by ID_process(), logsctpassoc(), and ProcessSctpMsg().

4.2.2.27 #define SN_LOCAL_TBL 0x01

assoc in local table

Definition at line 134 of file alias_sctp.h.

Referenced by AddSctpAssocLocal(), and RmSctpAssoc().

4.2.2.28 #define SN_NAT_PKT 0

Network Address Translate packet

Definition at line 150 of file alias_sctp.h.

Referenced by CL_process(), ID_process(), INa_process(), INi_process(), ProcessSctpMsg(), SctpAliasIn(), SctpAliasOut(), and UP_process().

4.2.2.29 #define SN_NULL_TBL 0x00

assoc in No table

Definition at line 136 of file alias_sctp.h.

Referenced by Sctp_PktParser().

4.2.2.30 #define SN_PARSE_ERROR_AS_MALLOC 2

Packet parsing error - assoc malloc

Definition at line 166 of file alias_sctp.h.

Referenced by Sctp_PktParser().

4.2.2.31 #define SN_PARSE_ERROR_CHHL 3

Packet parsing error - Chunk header len

Definition at line 167 of file alias_sctp.h.

Referenced by Sctp_PktParser().

4.2.2.32 #define SN_PARSE_ERROR_CHUNK 6

Packet parsing error - Chunk

Definition at line 170 of file alias_sctp.h.

Referenced by SctpPktParser().

4.2.2.33 #define SN_PARSE_ERROR_DIR 4

Packet parsing error - Direction

Definition at line 168 of file alias_sctp.h.

Referenced by SctpPktParser().

4.2.2.34 #define SN_PARSE_ERROR_IPSHL 1

Packet parsing error - IP and SCTP common header len

Definition at line 165 of file alias_sctp.h.

Referenced by SctpPktParser().

4.2.2.35 #define SN_PARSE_ERROR_VTAG 5

Packet parsing error - Vtag

Definition at line 169 of file alias_sctp.h.

Referenced by SctpPktParser().

4.2.2.36 #define SN_PARSE_OK 0

Packet parsing return results Packet parsed for SCTP messages

Definition at line 164 of file alias_sctp.h.

Referenced by SctpPktParser(), SctpAliasIn(), and SctpAliasOut().

4.2.2.37 #define SN_PROCESSING_ERROR 5

Packet processing error

Definition at line 158 of file alias_sctp.h.

Referenced by ProcessSctpMsg().

4.2.2.38 #define SN_REPLY_ABORT 2

Reply with ABORT to sender (don't forward it)

Definition at line 152 of file alias_sctp.h.

Referenced by ID_process(), SctpAliasIn(), and SctpAliasOut().

4.2.2.39 #define SN_REPLY_ERROR 3

Reply with ERROR to sender (don't forward it)

Definition at line 153 of file alias_sctp.h.

Referenced by ID_process(), SctpAliasIn(), and SctpAliasOut().

4.2.2.40 #define SN_RM 0x2000

Removing state

Definition at line 180 of file alias_sctp.h.

Referenced by CL_process(), INa_process(), INi_process(), logscpassoc(), ProcessSctpMsg(), SctpAliasIn(), SctpAliasOut(), and UP_process().

4.2.2.41 #define SN_SEND_ABORT 4

Send ABORT to destination

Definition at line 154 of file alias_sctp.h.

Referenced by INi_process(), SctpAliasIn(), and SctpAliasOut().

4.2.2.42 #define SN_SHUTDOWN 0

This variable defines how the NAT closes associations If SN_SHUTDOWN = 0, association is shutdown when the nat receives a ShutdownComplete If SN_SHUTDOWN > 0, the value represents the number of seconds to remain open after the receipt of a ShutdownComplete defines how NAT closes associations

Definition at line 122 of file alias_sctp.h.

Referenced by CL_process().

4.2.2.43 #define SN_TABLE_GLOBAL_SIZE 1001

Size of global link table

Definition at line 126 of file alias_sctp.h.

Referenced by AliasSctpInit(), logSctpGlobal(), and StartPointGlobal().

4.2.2.44 #define SN_TABLE_LOCAL_SIZE 1001

Size of local link table

Definition at line 125 of file alias_sctp.h.

Referenced by AliasSctpInit(), logSctpLocal(), and StartPointLocal().

4.2.2.45 #define SN_TO_GLOBAL 1

packet traveling from local to global

Definition at line 131 of file alias_sctp.h.

Referenced by ID_process(), INi_process(), logctpparse(), SCTP_PktParser(), and SctpAliasOut().

4.2.2.46 #define SN_TO_LOCAL 0

packet traveling from global to local

Definition at line 130 of file alias_sctp.h.

Referenced by ID_process(), INi_process(), logctpparse(), SCTP_PktParser(), and SctpAliasIn().

4.2.2.47 #define SN_U_T SN_CURTIME+SN_UT

state expiration time in seconds

Definition at line 185 of file alias_sctp.h.

Referenced by INa_process(), INi_process(), and UP_process().

4.2.2.48 #define SN_UP 0x0100

Association in UP state

Definition at line 178 of file alias_sctp.h.

Referenced by INa_process(), INi_process(), logctpassoc(), and ProcessSctpMsg().

4.2.2.49 #define SN_UT 600

Definition at line 182 of file alias_sctp.h.

4.2.2.50 #define SN_X_T SN_CURTIME+3

Definition at line 190 of file alias_sctp.h.

Referenced by CL_process().

Index

- `_SCTP_NAT_DEBUG`
 - `alias_sctp.h`, 39
- `AddSctpAssocGlobal`
 - `alias_sctp.c`, 17
- `AddSctpAssocLocal`
 - `alias_sctp.c`, 17
- `alias_sctp.c`, 13
 - `AddSctpAssocGlobal`, 17
 - `AddSctpAssocLocal`, 17
 - `AliasSctpInit`, 18
 - `AliasSctpTerm`, 18
 - `CL_process`, 19
 - `FindSctpGlobal`, 19
 - `FindSctpGlobalT`, 20
 - `FindSctpLocal`, 20
 - `FindSctpLocalT`, 21
 - `ID_process`, 21
 - `INa_process`, 22
 - `INi_process`, 23
 - `logsctpassoc`, 24
 - `logsctperror`, 24
 - `logSctpGlobal`, 25
 - `logSctpLocal`, 25
 - `logsctpparse`, 26
 - `logTimerQ`, 26
 - `ProcessSctpMsg`, 26
 - `ReplyAbortM`, 27
 - `ReplyErrorM`, 28
 - `RmSctpAssoc`, 28
 - `sctp_AddTimeOut`, 29
 - `sctp_CheckExp`, 29
 - `sctp_FirstChunkHdr`, 30
 - `sctp_NextChunkHdr`, 30
 - `SCTP_PktParser`, 30
 - `sctp_ResetTimeOut`, 31
 - `sctp_RmTimeOut`, 32
 - `SctpAliasIn`, 32
 - `SctpAliasLog`, 33
 - `SctpAliasOut`, 33
 - `SctpShowAliasStats`, 35
 - `sn_calloc`, 16
 - `sn_free`, 16
 - `sn_malloc`, 16
 - `StartPointGlobal`, 35
 - `StartPointLocal`, 35
 - `UP_process`, 35
- `alias_sctp.h`, 37
 - `_SCTP_NAT_DEBUG`, 39
 - `LINK_SCTP`, 39
 - `MAX_SCTP_TIMERQ`, 39
 - `SCTP_PACKED`, 40
 - `SCTP_UNUSED`, 40
 - `sctpAbort`, 40
 - `sctpAddIp`, 40
 - `sctpAddIpAck`, 40
 - `sctpInit`, 40
 - `sctpInitAck`, 40
 - `sctpOther`, 41
 - `sctpShutAck`, 41
 - `sctpShutComp`, 41
 - `SN_A_T`, 41
 - `SN_ADD_CLASH`, 41
 - `SN_ADD_OK`, 41
 - `SN_BOTH_TBL`, 41
 - `SN_C_T`, 41
 - `SN_CL`, 42
 - `SN_CURTIME`, 42
 - `SN_DROP_PKT`, 42
 - `SN_GLOBAL_TBL`, 42
 - `SN_I_T`, 42
 - `SN_ID`, 42
 - `SN_INa`, 42
 - `SN_INi`, 43
 - `SN_LOCAL_TBL`, 43
 - `SN_NAT_PKT`, 43
 - `SN_NULL_TBL`, 43
 - `SN_PARSE_ERROR_AS_MALLOC`, 43
 - `SN_PARSE_ERROR_CHHL`, 43
 - `SN_PARSE_ERROR_CHUNK`, 43
 - `SN_PARSE_ERROR_DIR`, 44
 - `SN_PARSE_ERROR_IPSHL`, 44
 - `SN_PARSE_ERROR_VTAG`, 44
 - `SN_PARSE_OK`, 44
 - `SN_PROCESSING_ERROR`, 44
 - `SN_REPLY_ABORT`, 44
 - `SN_REPLY_ERROR`, 44
 - `SN_RM`, 45
 - `SN_SEND_ABORT`, 45
 - `SN_SHUTDOWN`, 45

- SN_TABLE_GLOBAL_SIZE, 45
- SN_TABLE_LOCAL_SIZE, 45
- SN_TO_GLOBAL, 45
- SN_TO_LOCAL, 46
- SN_U_T, 46
- SN_UP, 46
- SN_UT, 46
- SN_X_T, 46
- AliasSctpInit
 - alias_sctp.c, 18
- AliasSctpTerm
 - alias_sctp.c, 18
- Asconf
 - sctpChunkOfInt, 11
- CL_process
 - alias_sctp.c, 19
- cur_loc
 - sctp_nat_timer, 10
- exp
 - sctp_nat_assoc, 7
- exp_loc
 - sctp_nat_assoc, 7
- FindSctpGlobal
 - alias_sctp.c, 19
- FindSctpGlobalT
 - alias_sctp.c, 20
- FindSctpLocal
 - alias_sctp.c, 20
- FindSctpLocalT
 - alias_sctp.c, 21
- g_port
 - sctp_nat_assoc, 6
- g_vtag
 - sctp_nat_assoc, 6
- ID_process
 - alias_sctp.c, 21
- INa_process
 - alias_sctp.c, 22
- INi_process
 - alias_sctp.c, 23
- Init
 - sctpChunkOfInt, 11
- InitAck
 - sctpChunkOfInt, 11
- ip_hdr
 - sctp_nat_msg, 8
- l_addr
 - sctp_nat_assoc, 6
- l_port
 - sctp_nat_assoc, 6
- l_vtag
 - sctp_nat_assoc, 6
- LINK_SCTP
 - alias_sctp.h, 39
- LIST_ENTRY
 - sctp_nat_assoc, 6
- LIST_HEAD
 - sctp_nat_timer, 10
- loc_time
 - sctp_nat_timer, 10
- logsctpassoc
 - alias_sctp.c, 24
- logsctperror
 - alias_sctp.c, 24
- logSctpGlobal
 - alias_sctp.c, 25
- logSctpLocal
 - alias_sctp.c, 25
- logsctpparse
 - alias_sctp.c, 26
- logTimerQ
 - alias_sctp.c, 26
- MAX_SCTP_TIMERQ
 - alias_sctp.h, 39
- msg
 - sctp_nat_msg, 8
- ProcessSctpMsg
 - alias_sctp.c, 26
- ReplyAbortM
 - alias_sctp.c, 27
- ReplyErrorM
 - alias_sctp.c, 28
- RmSctpAssoc
 - alias_sctp.c, 28
- sctp_AddTimeOut
 - alias_sctp.c, 29
- sctp_CheckExp
 - alias_sctp.c, 29
- sctp_FirstChunkHdr
 - alias_sctp.c, 30
- sctp_hdr
 - sctp_nat_msg, 8
- sctp_nat_assoc, 5
 - exp, 7
 - exp_loc, 7
 - g_port, 6
 - g_vtag, 6
 - l_addr, 6
 - l_port, 6

- l_vtag, 6
- LIST_ENTRY, 6
 - state, 7
 - TableRegister, 7
- sctp_nat_msg, 8
 - ip_hdr, 8
 - msg, 8
 - sctp_hdr, 8
 - sctpchnk, 9
- sctp_nat_timer, 10
 - cur_loc, 10
 - LIST_HEAD, 10
 - loc_time, 10
- sctp_NextChunkHdr
 - alias_sctp.c, 30
- SCTP_PACKED
 - alias_sctp.h, 40
- SCTP_PktParser
 - alias_sctp.c, 30
- sctp_ResetTimeout
 - alias_sctp.c, 31
- sctp_RmTimeout
 - alias_sctp.c, 32
- SCTP_UNUSED
 - alias_sctp.h, 40
- sctpAbort
 - alias_sctp.h, 40
- sctpAddIp
 - alias_sctp.h, 40
- sctpAddIpAck
 - alias_sctp.h, 40
- SctpAliasIn
 - alias_sctp.c, 32
- SctpAliasLog
 - alias_sctp.c, 33
- SctpAliasOut
 - alias_sctp.c, 33
- sctpchnk
 - sctp_nat_msg, 9
- sctpChunkOfInt, 11
 - Asconf, 11
 - Init, 11
 - InitAck, 11
- sctpInit
 - alias_sctp.h, 40
- sctpInitAck
 - alias_sctp.h, 40
- sctpOther
 - alias_sctp.h, 41
- SctpShowAliasStats
 - alias_sctp.c, 35
- sctpShutAck
 - alias_sctp.h, 41
- sctpShutComp
 - alias_sctp.h, 41
- SN_A_T
 - alias_sctp.h, 41
- SN_ADD_CLASH
 - alias_sctp.h, 41
- SN_ADD_OK
 - alias_sctp.h, 41
- SN_BOTH_TBL
 - alias_sctp.h, 41
- SN_C_T
 - alias_sctp.h, 41
- sn_calloc
 - alias_sctp.c, 16
- SN_CL
 - alias_sctp.h, 42
- SN_CURTIME
 - alias_sctp.h, 42
- SN_DROP_PKT
 - alias_sctp.h, 42
- sn_free
 - alias_sctp.c, 16
- SN_GLOBAL_TBL
 - alias_sctp.h, 42
- SN_I_T
 - alias_sctp.h, 42
- SN_ID
 - alias_sctp.h, 42
- SN_INa
 - alias_sctp.h, 42
- SN_INi
 - alias_sctp.h, 43
- SN_LOCAL_TBL
 - alias_sctp.h, 43
- sn_malloc
 - alias_sctp.c, 16
- SN_NAT_PKT
 - alias_sctp.h, 43
- SN_NULL_TBL
 - alias_sctp.h, 43
- SN_PARSE_ERROR_AS_MALLOC
 - alias_sctp.h, 43
- SN_PARSE_ERROR_CHHL
 - alias_sctp.h, 43
- SN_PARSE_ERROR_CHUNK
 - alias_sctp.h, 43
- SN_PARSE_ERROR_DIR
 - alias_sctp.h, 44
- SN_PARSE_ERROR_IPSHL
 - alias_sctp.h, 44
- SN_PARSE_ERROR_VTAG
 - alias_sctp.h, 44
- SN_PARSE_OK
 - alias_sctp.h, 44
- SN_PROCESSING_ERROR

- alias_sctp.h, [44](#)
- SN_REPLY_ABORT
 - alias_sctp.h, [44](#)
- SN_REPLY_ERROR
 - alias_sctp.h, [44](#)
- SN_RM
 - alias_sctp.h, [45](#)
- SN_SEND_ABORT
 - alias_sctp.h, [45](#)
- SN_SHUTDOWN
 - alias_sctp.h, [45](#)
- SN_TABLE_GLOBAL_SIZE
 - alias_sctp.h, [45](#)
- SN_TABLE_LOCAL_SIZE
 - alias_sctp.h, [45](#)
- SN_TO_GLOBAL
 - alias_sctp.h, [45](#)
- SN_TO_LOCAL
 - alias_sctp.h, [46](#)
- SN_U_T
 - alias_sctp.h, [46](#)
- SN_UP
 - alias_sctp.h, [46](#)
- SN_UT
 - alias_sctp.h, [46](#)
- SN_X_T
 - alias_sctp.h, [46](#)
- StartPointGlobal
 - alias_sctp.c, [35](#)
- StartPointLocal
 - alias_sctp.c, [35](#)
- state
 - sctp_nat_assoc, [7](#)
- TableRegister
 - sctp_nat_assoc, [7](#)
- UP_process
 - alias_sctp.c, [35](#)