

|  |
| --- |
|  |

Switch Abstraction Interface

Change Proposal

|  |  |
| --- | --- |
| **Title** | **Mirroring API’s** |
| **Authors** | **DELL** |
| **Status** | **In Review** |
| **Type** | **Standards Track** |
| **Created** | **02/05/2015** |
| **SAI-Version** | **V0.9.2** |

**Contents**

[List of Changes i](#_Toc411855743)

[1 Overview 1](#_Toc411855744)

[2 Specification 1](#_Toc411855745)

[2.1 Changes to sai.h 1](#_Toc411855746)

[2.2 Changes to saitypes.h 1](#_Toc411855747)

[2.3 Changes to saiport.h 1](#_Toc411855748)

[2.4 New definitions in saimirror.h 2](#_Toc411855749)

# List of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Changes | Name | Date |
| 0.9.2 | Proposal for Mirror – Version 1 |  | 2/5/15 |
| 0.9.2 | Version 2. Updated with the following changes   * Made the Mirror enable/Disable a port based attribute * Replaced all instances of SPAN with Mirroring * Added a new field for GRE Protocol Type |  | 2/16/15 |

License

© 2014 Microsoft Corporation, Dell Inc., Facebook, Inc, Broadcom Corporation, Intel Corporation, Mellanox Technologies Ltd.

As of September 9, 2014, the following persons or entities have made this Specification available under the Open Web Foundation Final Specification Agreement (OWFa 1.0), which is available at <http://www.openwebfoundation.org/legal/the-owf-1-0-agreements/owfa-1-0>

Microsoft Corporation, Dell Inc., Facebook, Inc, Intel Corporation, Mellanox Technologies Ltd.

You can review the signed copies of the Open Web Foundation Agreement Version 1.0 for this Specification at <http://opencompute.org/licensing/>, which may also include additional parties to those listed above.

Your use of this Specification may be subject to other third party rights. THIS SPECIFICATION IS PROVIDED "AS IS." The contributors expressly disclaim any warranties (express, implied, or otherwise), including implied warranties of merchantability, noninfringement, fitness for a particular purpose, or title, related to the Specification. The entire risk as to implementing or otherwise using the Specification is assumed by the Specification implementer and user. IN NO EVENT WILL ANY PARTY BE LIABLE TO ANY OTHER PARTY FOR LOST PROFITS OR ANY FORM OF INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION OR ITS GOVERNING AGREEMENT, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND WHETHER OR NOT THE OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE FOLLOWING IS A LIST OF MERELY REFERENCED TECHNOLOGY: Microprocessor technology, semiconductor manufacturing technology, operating system technology (including without limitation networking operating system technology), emulation technology, graphics technology, video technology, integrated circuit packaging technology and the like, compiler technologies, object oriented technology, optical/RF communications technology including chip I/O and driver technology, bus technology, memory chip technology (including, without limitation, NAND memory, NOR memory, resistive RAM (RRAM), seek scan probe (SSP) memory, nonvolatile memory (including without limitation, memory based on chalcogenide materials, phase change memory (PCM), one or more stacked layers of memory cells, embedded PCM memories, non-volatile cache memory, solid state drives, SRAM, embedded DRAM, ferro-electric memory, and polymer memory)) and/or health-related and medical technology. IMPLEMENTATION OF THESE TECHNOLOGIES MAY BE SUBJECT TO THEIR OWN LEGAL TERMS.

# Overview

Proposal for saimirror.h interface

# Specification

**Mandatory**

* Implements port mirroring management functions like
* Creating a mirror session
* Specifying the type of mirroring
* Adding source ports to the session
* Removing the source ports from the session
* Destroying the session
* Specifying various parameters to create a session

## Changes to sai.h

typedef enum \_sai\_api\_t {

SAI\_API\_MIRROR= 14, /\* sai\_mirror\_api\_t \*/

} sai\_api\_t;

## Changes to saitypes.h

#define UINT32 sai\_mirror\_session\_id\_t

**Changes to existing sai\_attribute\_value\_t definition**

typedef union {

sai\_mirror\_session\_list\_t mirror\_session\_list ;

} sai\_attribute\_value\_t ;

typedef struct \_sai\_mirror\_session\_list\_t {

uint32\_t session\_count;

sai\_mirror\_session\_id\_t \*session\_id;

} **sai\_mirror\_session\_list\_t**;

## Changes to saiport.h

typedef enum \_sai\_port\_attr\_t

{

/\* READ-WRITE \*/

/\* Mirror session [sai\_mirror\_session\_list\_t] \*/

SAI\_PORT\_ATTR\_INGRESS\_MIRROR\_SESSION

/\* READ-WRITE \*/

/\* Mirror session [sai\_mirror\_session\_list\_t] \*/

SAI\_PORT\_ATTR\_EGRESS\_MIRROR\_SESSION

} sai\_port\_attr\_t

## New definitions in saimirror.h

/\* Specify the type of mirroring \*/

typedef enum \_sai\_mirror\_type\_t {

/\* Local span \*/

SAI\_MIRROR\_TYPE\_LOCAL = 1,

/\* Remote span \*/

SAI\_MIRROR\_TYPE\_REMOTE,

/\* Enhanced Remote span \*/

SAI\_MIRROR\_TYPE\_ENHANCED\_REMOTE,

} **sai\_mirror\_type\_t**;

typedef enum \_sai\_erspan\_encapsulation\_type\_t

{

/\* L2 Tunnel Encapsulation \*/

SAI\_MIRROR\_L2\_TUNNEL,

/\* L3 Tunnel Encapsulation \*/

SAI\_MIRROR\_L3\_GRE\_TUNNEL,

} **sai\_erspan\_encapsulation\_type\_t**;

typedef enum \_sai\_mirror\_session\_attr\_t {

/\* CREATE + READ-ONLY \*/

/\* Destination/Analyser/Monitor Port - mandatory for create [sai\_port\_id\_t]\*/

SAI\_MIRROR\_SESSION\_ATTR\_MONITOR\_PORT,

/\* CREATE + READ-ONLY \*/

/\* Class-of-Service (Traffic Class) - mandatory for create [uint8\_t] \*/

SAI\_MIRROR\_SESSION\_ATTR\_COS,

/\* CREATE + READ-ONLY \*/

/\* Valid for RSPAN and ERSPAN

\* L2 header TPID if vlanId is not zero - mandatory for create [uint16\_t]\*/

SAI\_MIRROR\_SESSION\_ATTR\_VLAN\_TPID,

/\* CREATE + READ-ONLY \*/

/\* Valid for RSPAN and ERSPAN L2 header VlanId - mandatory for create [sai\_vlan\_id\_t]\*/

SAI\_MIRROR\_SESSION\_ATTR\_VLAN\_ID,

/\* CREATE + READ-ONLY \*/

/\* Valid for RSPAN and ERSPAN packet priority - mandatory for create [uint8\_t] \*/

SAI\_MIRROR\_SESSION\_ATTR\_VLAN\_PRI,

/\* All attributes below are Valid only for ERSPAN \*/

/\* CREATE + READ-ONLY \*/

/\* Encapsulation type - sai\_erspan\_encapsulation\_type\_t - mandatory for create \*/

SAI\_MIRROR\_SESSION\_ATTR\_ENCAP\_TYPE,

/\* CREATE + READ-ONLY \*/

/\* tunnel IP header version - mandatory for create [uint8\_t]\*/

SAI\_MIRROR\_SESSION\_ATTR\_IPHDR\_VERSION,

/\* CREATE + READ-ONLY \*/

/\* tunnel header TOS - mandatory for create [uint8\_t]\*/

SAI\_MIRROR\_SESSION\_ATTR\_TOS,

/\* CREATE + READ-ONLY \*/

/\* tunnel header TTL - mandatory for create [uint8\_t]\*/

SAI\_MIRROR\_SESSION\_ATTR\_TTL,

/\* CREATE + READ-ONLY \*/

/\* tunnel source IP - mandatory for create [sai\_ip\_address\_t] \*/

SAI\_MIRROR\_SESSION\_ATTR\_SRC\_IP\_ADDRESS,

/\* CREATE + READ-ONLY \*/

/\* tunnel destination IP - mandatory for create [sai\_ip\_address\_t] \*/

SAI\_MIRROR\_SESSION\_ATTR\_DST\_IP\_ADDRESS,

/\* CREATE + READ-ONLY \*/

/\* L2 source MAC address - mandatory for create [sai\_mac\_t] \*/

SAI\_MIRROR\_SESSION\_ATTR\_SRC\_MAC\_ADDRESS,

/\* CREATE + READ-ONLY \*/

/\* L2 destination MAC address - mandatory for create [sai\_mac\_t] \*/

SAI\_MIRROR\_SESSION\_ATTR\_DST\_MAC\_ADDRESS,

/\* CREATE + READ-ONLY \*/

/\* GRE protocol Id – mandatory for create [uint16\_t] \*/

SAI\_MIRROR\_SESSION\_ATTR\_GRE\_PROTOCOL\_TYPE,

} **sai\_mirror\_session\_attr\_t**;

/\*

\* Routine Description:

\* Create mirror session.

\*

\* Arguments:

\* [inout] session\_id - port mirror session

\* [in] mirror\_type - SPAN/RSPAN/ERSPAN

\* [in] attr\_count - number of attributes

\* [in] attr\_list - array of attributes

\*

\* Return Values:

\* SAI\_STATUS\_SUCCESS on success

\* Failure status code on error

\*/

typedef sai\_status\_t (\*sai\_create\_mirror\_session\_fn)(

\_Inout\_ sai\_mirror\_session\_id\_t \*session\_id,

\_In\_ sai\_mirror\_type\_t mirror\_type,

\_In\_ uint32\_t attr\_count,

\_In\_ const sai\_attribute\_t \*attr\_list);

/\*

\* Routine Description:

\* Destroy Mirror session.

\*

\* Arguments:

\* [in] session\_id - port mirror session

\*

\* Return Values:

\* SAI\_STATUS\_SUCCESS on success

\* Failure status code on error

\*/

typedef sai\_status\_t (\*sai\_remove\_mirror\_session\_fn)(

\_In\_ sai\_mirror\_session\_id\_t session\_id);

/\*

\* Routine Description:

\* Set Mirror session attribute.

\*

\* Arguments:

\* [in] session\_id - session\_id to set the attributes

\* [in] attr – attribute value

\*

\* Return Values:

\* SAI\_STATUS\_SUCCESS on success

\* Failure status code on error

\*/

typedef sai\_status\_t (\*sai\_set\_mirror\_session\_attribute\_fn)(

\_In\_ sai\_mirror\_session\_id\_t session\_id,

\_In\_ const sai\_attribute\_t \*attr);

/\*

\* Routine Description:

\* Get mirroring session attribute.

\*

\* Arguments:

\* [in] session\_id - session\_id to retrieve the attribute

\* [in] attr\_count - number of attributes

\* [inout] attr\_list - array of attributes

\*

\* Return Values:

\* SAI\_STATUS\_SUCCESS on success

\* Failure status code on error

\*/

typedef sai\_status\_t (\*sai\_get\_mirror\_session\_attribute\_fn)(

\_In\_ sai\_mirror\_session\_id\_t session\_id,

\_In\_ uint32\_t attr\_count,

\_Inout\_ sai\_attribute\_t \*attr\_list);

/\* MIRROR method table retrieved with sai\_api\_query() \*/

typedef struct \_sai\_mirror\_api\_t

{

sai\_create\_mirror\_session\_fn create\_mirror\_session;

sai\_delete\_mirror\_session\_fn delete\_mirror\_session;

sai\_set\_mirror\_session\_attribute\_fn set\_mirror\_session\_attribute;

sai\_get\_mirror\_session\_attribute\_fn get\_mirror\_session\_attribute;

} sai\_mirror\_api\_t;