



Switch Abstraction Interface (SAI) Specification for CSIG

v1.0

October 28, 2026

Copyright © 2026 Ultra Ethernet Consortium.

The Ultra Ethernet Consortium™ (UEC) is a consensus-based standards organization operating under the Linux Foundation. Its members collaborate openly to define and promote high-performance Ethernet technologies for modern computing environments. This specification ("Specification") contains Approved Deliverables of UEC as such term is defined in the UEC Charter.

This document is made available under the terms of the Creative Commons Attribution-NoDerivatives 4.0 International License (CC BY-ND 4.0). As stated in the license, you may copy and redistribute the document, provided that attribution is given to the Ultra Ethernet Consortium. However, if you create a derivative work from the material, you may not distribute that work. A copy of the license is available at <https://creativecommons.org/licenses/by-nd/4.0/>.

This Specification is provided on an "AS IS" basis. UEC and The Linux Foundation (the "Parties") make no warranties of any kind, either express or implied, including without limitation warranties of merchantability, fitness for a particular purpose or noninfringement of any third party intellectual property rights, or compliance with applicable laws. Use of the information in this Specification is at your own risk. The entire risk as to the results and performance of this Specification is assumed by the user. In no event will the Parties (as defined above) 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 document, whether based on breach of contract, tort (including negligence), or otherwise, and whether or not the recipient of this Specification has been advised of the possibility of such damage.

This Specification may include references to technologies for which patent rights might be claimed. UEC does not make any representation regarding the existence or status of any such rights. Implementers of this Specification are solely responsible for obtaining any necessary licenses or permissions from the appropriate rights holders.

UEC reserves the right to adopt any changes or alterations to this Specification as it deems necessary or appropriate.

Ultra Ethernet™, Ultra Ethernet Consortium™, and the Ultra Ethernet Consortium logo are the unregistered trademarks of Ultra Ethernet Consortium in the United States and other countries. All rights reserved. Other brands, names or logos contained herein are the property of their respective owners.

For more information about the Ultra Ethernet Consortium, visit: <https://ultraethernet.org>.

Contents

1	Introduction	7
1.1	Terminology	8
1.2	Overview	8
1.2.1	csig.d processing	10
1.3	SAI Specification	12
1.3.1	TAM INT presence type	12
1.3.2	TAM INT Type	13
1.3.3	New enum for CSIG Signal Types	13
1.3.4	TAM INT BW Time Interval	13
1.3.5	TAM INT Object	13
1.3.6	ACL Match Fields	14
1.3.7	ACL Actions	15
1.3.8	New ACL data type	15
1.3.9	Switch and Port Object Attributes	15
1.3.10	CSIG Stats	15
1.3.11	Capability Query	16
1.4	Normative References	16

Tables

Table 1-1 - CSIG Tag Fields	8
Table 1-2 - 4-Byte Compact CSIG Tag Field Definitions	9
Table 1-3 - 8-Byte Wide CSIG Tag Field Definitions	9
Table 1-4 - Signal Types for Compact and Wide Tag	10
Table 1-5 - TAM INT Presence Type	13
Table 1-6 - TAM INT Type.....	13
Table 1-7 - Switch Level Read Only Attribute For Time Interval Values	13
Table 1-8 - Time Interval Configuration	13
Table 1-9 - TAM INT Object Attributes	14
Table 1-10 - SAI_OBJECT_TYPE_ACL_ENTRY Enhancements.....	14
Table 1-11 - ACL CSIG Signal Data Type.....	15
Table 1-12 - Switch and Port Attributes	15

Figures

Figure 1-1 - CSIG Domain with 4-Byte and 8-Byte CSIG tag.....	7
Figure 1-3 - 4-Byte Compact CSIG Tag.....	9
Figure 1-2 - 8-Byte Wide CSIG Tag	9
Figure 1-4 - CSIG Tag Position in L2 Ethernet Headers	11
Figure 1-5 - Example of CSIG Endpoint and Fabric Processing	11
Figure 1-6 - Overview of SAI objects for CSIG	12

Acknowledgments

This specification is developed under the Process, Procedures, and Guidelines of the Ultra Ethernet Consortium (UEC). The organization would like to thank all that are involved. A special acknowledgment goes out to the following individuals for their leadership, authoring, and technical contributions to this specification.

Steering Committee at Time of Publication

Role	Name	Affiliation
Chair	J Metz	AMD
Vice Chair	Abdul Kabbani	Microsoft
TAC Chair	Tom Emmons	Arista
TAC Vice Chair	Brad Matthews	Marvell

Working Group Leadership at Time of Publication

Working Group	Name	Affiliation
Physical Layer WG	Jannik Hammel Nielsen	Huawei
	Adee Ran	Cisco
Link Layer WG	Bob Alverson	HPE
	Yan Zhuang	Huawei
Transport Layer WG	Torsten Hoefler	Microsoft
	Karen Schramm	Broadcom
Software Layer WG	Rip Sohan	AMD
	Eric Spada	Broadcom
Management WG	Ravindranath Kanakarajan	Marvell
	Hemal Shah	Broadcom
Compliance & Test WG	Dave Estes	Viavi Solutions
	Cliff Tavares	Keysight
Performance & Debug WG	Rajalaxmi Angadi	Intel
	Nathan Hanford	LLNL
Storage WG	Phil Cayton	Intel
	Claudio DeSanti	Dell

Specification Development

Role	Name	Affiliation
Author	Jai Kumar	Broadcom
Contributor	Ravindranath Kanakarajan	Marvell
Contributor	Rupa Budhia	Marvell
Editor	Paul Congdon	Linux Foundation
Program Manager	Rob Craig	Linux Foundation

1 Introduction

Congestion Signaling (CSIG) is a simple and efficient network telemetry method that collects fine-grained signals that describe the network fabric's congestion state.

The core principle of CSIG's design is the aggregation of congestion state information collected at switches along a packet's path. For many useful telemetry signals employed in congestion control and performance analysis, the key metric is the result of a simple aggregation function applied to the congestion state values.

For instance, with several signal types, we are primarily interested in the congestion state at only one specific switch along the path. This "switch of interest" is defined as the one where the signal's value represents an extremum (either the maximum or minimum) among all switches along that path.

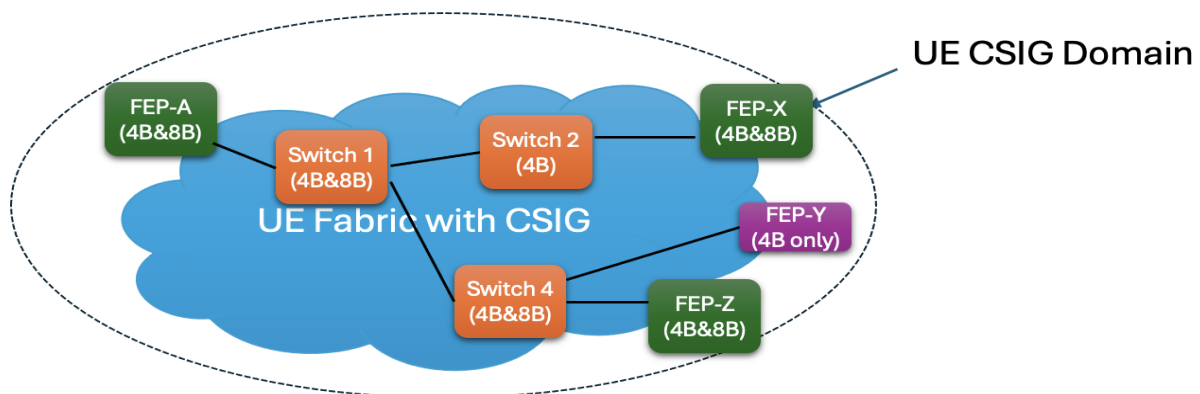


Figure 1-1 - CSIG Domain with 4-Byte and 8-Byte CSIG tag

Following logical functions can be performed by the UE fabric element in a CSIG domain:

- CSIG-Init()
 - Only the Fabric End Point (FEP) executes the initialization function, which involves inserting the CSIG tag into the packet
- CSIG-Transit()
 - UE Fabric switch element performs the transit function, which may include *CSIG-Compare-and-Update()* or *Strip-and-Forward()* of the CSIG tag
- CSIG-Term()
 - While the FEP typically terminates the CSIG tag, the switch also has the capability to perform this termination.
- CSIG-Strip-and-Forward()
 - The switch handles terminating the CSIG tag by performing the *CSIG-Strip-and-Forward()* function when the peering switch does not support CSIG
- CSIG-Reflect()

- The receiving FEP performs the reflection, where the CSIG tag is reflected in the UET Reflection Header for sender-based congestion control

1.1 Terminology

Term	Description
CSIG	Congestion Signaling
Locator Metadata	Contextual information stored in a CSIG tag to identify where in the network that tag was updated
Monitoring Entity	Entity responsible for collecting telemetry captured via CSIG and/or other methods, provisioned to scale to report arrival rates
Signal Type and Value	In-network telemetry information of interest
SOP	Start of Packet
VLAN tag	An IEEE 802.1Q-2022 clause 9-compliant VLAN tag [802.1Q-2022]
FEP	Fabric End Point typically a NIC
TAM	Telemetry And Monitoring. A SAI object for anything telemetry.
INT	In-band Network Telemetry

1.2 Overview

Congestion Signaling (CSIG) is a UE specification for collecting telemetry data from the fabric elements. CSIG specification defines a new protocol header with encoding to indicate what type of signal data to collect.

This CSIG tag data can be used for

- Congestion Control Algorithms like NSCC, RCCC and for congestion control frameworks like PCM.
- Telemetry purposes (out of scope for the UE specification).

There are two types of CSIG tags defined

- 4-byte compact tag
- 8-byte wide tag

The CSIG tag is identified by an EtherType code point

- The 4-byte compact tag is identified using its own EtherType code point.
- The 8-byte wide tag is identified using its own EtherType code point.

The CSIG tag contains the fields defined in Table 1-1. The differences between the 4-Byte and 8-Byte CSIG tags are shown in Table 1-2 and Table 1-3.

Table 1-1 - CSIG Tag Fields

Field	Field Name	Description
t	Signal type	Specifies the kind of telemetry or signal data to collect
s	Signal value	The quantized signal value
lm	Locator metadata	An indication of which switch hop modified the signal value

Field	Field Name	Description
d	Do not modify	Set by a switch to prevent other switches from updating the tag (e.g., after trimming a packet)
r	Reserved	

Table 1-2 - 4-Byte Compact CSIG Tag Field Definitions

Bit Offset	Width (in bits)	Field Name	Comments
0-15	16	csig_tpid_cpct	New Ethertype allocated by IEEE
16-18	3	t	Type of Signal
19	1	r	Reserved
20-24	5	s	Quantized Signal Value
25-30	6	lm	Locator Metadata
31	1	d	Do Not Modify CSIG Tag

Table 1-3 - 8-Byte Wide CSIG Tag Field Definitions

Bit Offset	Width (in bits)	Field Name	Comments
0-15	16	csig_tpid_wide	New Ethertype allocated by IEEE
16-30	15	lm	Locator Metadata
31	1	d	Do Not Modify CSIG Tag
32-35	4	t	Signal Type
36-55	20	s	Quantized Signal Value
56-63	8	r	Reserved

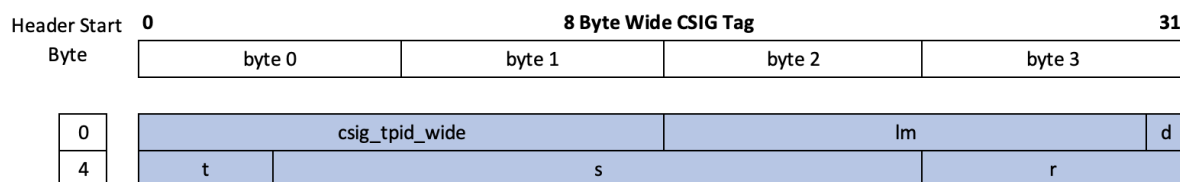


Figure 1-3 - 8-Byte Wide CSIG Tag

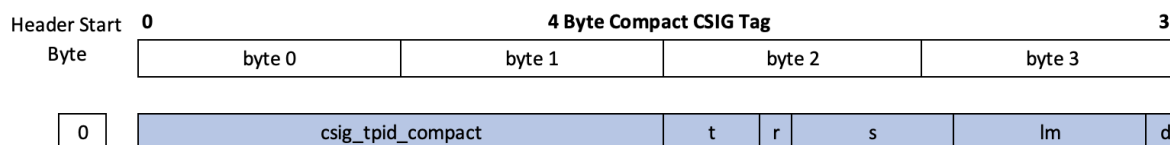


Figure 1-2 - 4-Byte Compact CSIG Tag

The signals defined in Table 1-4 are supported by both compact and wide tags.

Table 1-4 - Signal Types for Compact and Wide Tag

t	Signal	Profile	Aggregation Function	Comments
0	ABW	base	min	Available bandwidth per port
1	ABW/C	base	min	Relative available bandwidth per port
2	Delay	base	max	Per-hop delay
3	nQD	extended	max	Queue depth normalized by port speed

1.2.1 csig.d processing

The **csig.d** bit is set by the switch that performs the trimming operation on the ingress packet. Trimming typically occurs when a queue experiences congestion, leading to a packet drop.

The ability to update the **csig.d** bit is controlled by an associated policy configuration.

A CSIG compliant switch may not natively support the ability to set the **csig.d** bit. This capability (or lack thereof) MUST be explicitly exposed via a capability query mechanism.

The default behavior for packets with the **csig.d** bit set is to not update the CSIG tag, unless a specific policy is configured to override the default behavior.

1.2.1.1 Quantization

The compact and wide tags specify different methods of quantization.

The compact tag supports 32 fully configurable range buckets. In contrast, the wide tag supports one million buckets, which MUST be configured algorithmically for minimum and maximum ranges.

While the specific quantization algorithm is outside the scope of this specification, it utilizes a base value and a step value. The base value defines the starting point for observing a signal value, and the step value defines the width of the bucket. For the wide tag, the width of the bucket remains constant.

For ease of hardware implementation, both the base value and the step value MUST be powers of two.

1.2.1.2 Time interval for ABW and ABW/C signal value computation

The CSIG specification requires a consistent time interval, 't', to be configured across all switches within the fabric. This interval is specifically used to measure the amount of traffic in bytes. The resulting observed value is used in the subsequent computation of both the Available Bandwidth (ABW) and the ABW/C signal values.

For the detailed formula used in the ABW computation, please refer to the CSIG section of the UE Specification v1.1.

1.2.1.3 CSIG Tag Layout

Figure 1-4 shows the position of an Ethernet frame's CSIG tag among the fields of the frame's L2 Ethernet and IEEE 802.1Q headers. In these examples, CSIG-ETYPE refers to the 2-byte TPID EtherType value for either a compact or wide CSIG tag, as appropriate, and CSIG-DATA refers to the remaining two or six bytes of the compact or wide CSIG tag, respectively.

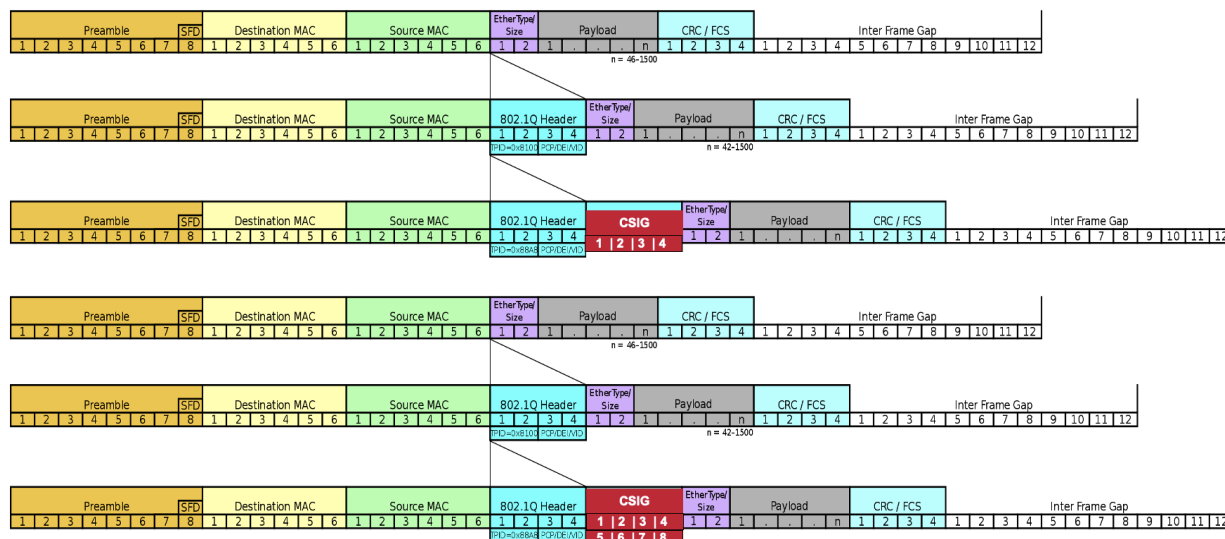


Figure 1-4 - CSIG Tag Position in L2 Ethernet Headers

1.2.1.4 Life of a CSIG Packet

This section uses a simple example, shown in Figure 1-5, to demonstrate how CSIG telemetry operates on endpoints and within the network fabric. The example depicts a high-level schematic of a CSIG-tagged packet, showing both the CSIG tag carried on a packet's forward path and the reflection of the signal from the destination endpoint back to the source endpoint in an L4 transport header. The example further shows how switches update the signal value field at each hop for three signal types—max(Delay), min(ABW), and min(ABW/C)—and the distinct bottleneck locations for each of these signal types along the path. The max(Delay) signal type finds the delay bottleneck along a path. It conditionally updates the queueing delay at each hop using the maximum function, where a switch overwrites the delay signal value in the CSIG tag with its own local delay value if that value exceeds the value in the CSIG tag. The min(ABW) signal type finds the available bandwidth bottleneck along a path. It aggregates

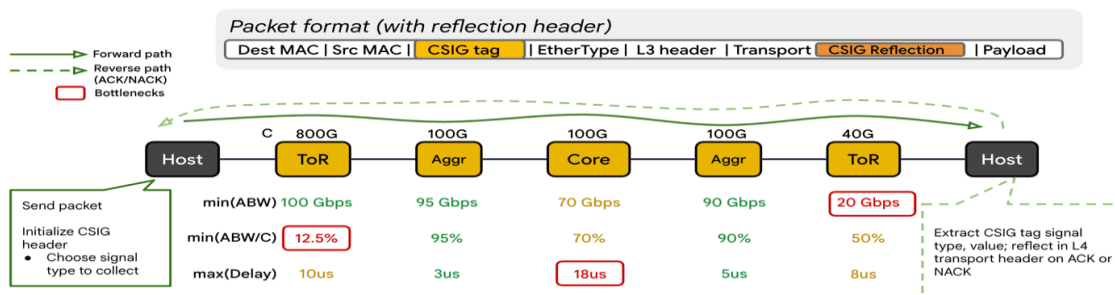


Figure 1-5 - Example of CSIG Endpoint and Fabric Processing

the available bandwidth at each hop using the minimum function, where a switch overwrites the available bandwidth value in the CSIG tag with its own local available bandwidth value if that value is less than the value in the CSIG tag. The min(ABW/C) signal type does the same as the min(ABW) signal type but for each hop's relative available bandwidth, normalized by output port capacity.

1.3 SAI Specification

CSIG tagged packets do not alter the forwarding behavior of the packet. Forwarding continues to rely on the standard mechanisms, such as the IP address or the UE UFH headers (see section 5.4 of UE Specification v1.1). Consequently, no changes are required within the SAI logical pipeline to support CSIG packets.

The existing SAI infrastructure leverages the Telemetry and Monitoring (TAM) SAIobject. The TAM framework currently supports features such as in-band network telemetry, streaming telemetry, events, transport for reporting, and various reporting formats. This specification proposes enhancing the existing TAM framework to include support for CSIG.

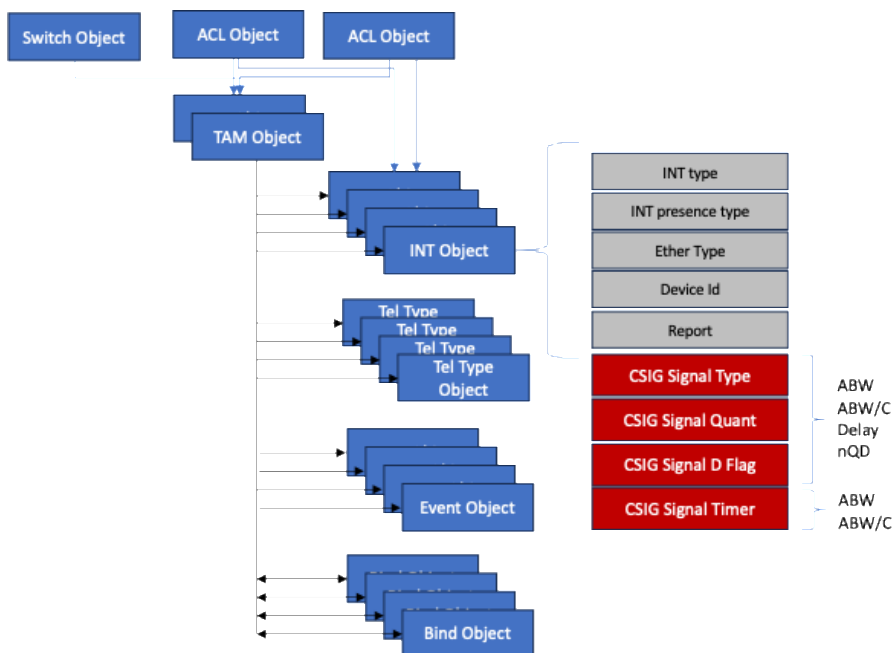


Figure 1-6 - Overview of SAI objects for CSIG

1.3.1 TAM INT presence type

The *sai_tam_int_presence_type_t* enum indicates how a given INT packet is identified.

A CSIG tag is identified using a new EtherType value. To accommodate this change, the TAM INT presence enumeration is updated with a new enum value for EtherType.

Table 1-5 - TAM INT Presence Type

typedef sai_tam_int_presence_type_t	Description
SAI_TAM_INT_PRESENCE_TYPE_ETH_TYPE	INT presence type is indicated by EtherType value in L2 header

1.3.2 TAM INT Type

The *sai_tam_int_type_t* enum specifies types of INT technologies supported by the SAI specification. This enum is enhanced with two new enum values, one for compact tag and other for wide tag.

Table 1-6 - TAM INT Type

typedef sai_tam_int_type_t	Description
SAI_TAM_INT_TYPE_CSIG_COMPACT	INT type Congestion Signaling 4 byte compact tag
SAI_TAM_INT_TYPE_CSIG_WIDE	INT type Congestion Signaling 8 byte wide tag

1.3.3 New enum for CSIG Signal Types

A new enum is introduced to specify the CSIG signal as part of the ACL infrastructure enhancement. This enum is referenced by the attribute in TAM INT object to configure a CSIG signal type.

1.3.4 TAM INT BW Time Interval

The time interval is configured for both the available bandwidth and normalized available bandwidth computations. To support this, a new read only enum is defined at a switch level to query all the supported time intervals by an implementation. Subsequently device can be configured with one of the read values using the new attribute. All the time interval values are in microseconds.

Table 1-7 - Switch Level Read Only Attribute For Time Interval Values

typedef sai_switch_attr_t	Description
SAI_SWITCH_ATTR_CSIG_BW_TIME_INTERVAL_IN_USEC	CSIG Bandwidth time intervals supported in microseconds valid only for available and relative available bandwidth signal type

Table 1-8 - Time Interval Configuration

typedef sai_tam_int_attr_t	Description
SAI_TAM_INT_ATTR_BW_TIME_INTERVAL	Time interval for bandwidth computation in microseconds

1.3.5 TAM INT Object

The TAM INT object serves as the primary container for various in-band network telemetry (INT) technologies, including P4-INT, IFA, and IOAM. It is responsible for enumerating all supported INT types and defining their respective identification methods. Furthermore, this object specifies all necessary attributes that are applicable across all or specific INT types.

This object is enhanced to support the new CSIG INT type, along with its associated identification method. New attributes are integrated to specify the signals, quantization method, and time intervals required for signal value computation.

Table 1-9 - TAM INT Object Attributes

Attribute	Type	Flags	Default	Description
SAI_TAM_INT_ATTR_ETH_TYPE_CODE_POINT	sai_uint16_t	MANDATORY_ON_CREATE CREATE_ONLY	n/a	Ethernet code point value that indicates presence of CSIG tag in a packet. Note that 4-byte and 8-byte tag have two distinct EtherType code points.
SAI_TAM_INT_ATTR_CSIG_SIGNAL_TYPE	sai_csig_signal_type_t	CREATE_AND_SET	SAI_CSIG_SIGNAL_TYPE_ABW	CSIG signal type
SAI_TAM_INT_ATTR_BW_TIME_INTERVAL	sai_uint32_t	CREATE_AND_SET	128 microseconds	Time interval for bandwidth computation in microseconds
SAI_TAM_INT_ATTR_QUANT_BANDS	sai_uint32_t	CREATE_AND_SET	32	Number of quantization bands. Compact tag supports maximum of 32 bands and wide tag supports maximum of 1048576 bands. Valid for both compact and wide tags
SAI_TAM_INT_ATTR_QUANT_BAND_ID	sai_u16_list_t	CREATE_AND_SET	empty	Quantization bands identifier. Number of identifiers specified MUST match the number of quantization bands configured for compact tag
SAI_TAM_INT_ATTR_QUANT_BAND_UINT64_RANGE_LIST	sai_u64_range_list_t	CREATE_AND_SET	empty	Quantization band range values for compact tag
SAI_TAM_INT_ATTR_WIDE_QUANT_BASE_VALUE	sai_uint32_t	CREATE_AND_SET	0	Base value for wide tag quantization. MUST be 0 or power of 2
SAI_TAM_INT_ATTR_WIDE_QUANT_STEP_VALUE	sai_uint32_t	CREATE_AND_SET	1	Step value for wide tag quantization. MUST be power of 2
SAI_TAM_INT_ATTR_CSIG_D_BIT	bool	CREATE_AND_SET	FALSE	Update the D bit indicating the packet is trimmed. DSCP value configured for TRIM packets comes from the TRIM configuration

1.3.6 ACL Match Fields

The ACL infrastructure supports matching specific fields within a packet and is enhanced to include CSIG tag fields.

A new match field for compact and wide tag EtherType and signal type is introduced.

Table 1-10 - SAI_OBJECT_TYPE_ACL_ENTRY Enhancements

Attribute	Type	Flags	Default	Description
SAI_ACL_ENTRY_ATTR_FIELD_CSIG_COMPACT_ETHER_TYPE	sai_uint16_t	CREATE_AND_SET	disabled	CSIG Compact Ether Type
SAI_ACL_ENTRY_ATTR_FIELD_CSIG_WIDE_ETHER_TYPE	sai_uint16_t	CREATE_AND_SET	disabled	CSIG Wide Ether Type
SAI_ACL_ENTRY_ATTR_FIELD_CSIG_SIGNAL_TYPE	sai_csig_signal_type_t	CREATE_AND_SET	disabled	CSIG Signal Type

Attribute	Type	Flags	Default	Description
SAI_ACL_ENTRY_ATTR_FIELD_CSIG_D_BIT	sai_uint8_t	CREATE_AND_SET	disabled	csig.d bit in the CSIG header

1.3.7 ACL Actions

This specification does not introduce any new ACL actions. However, there is a scenario where a CSIG tag with the **csig.d** bit ‘set’ might necessitate suppressing the compare and update operations. This can be accomplished by implementing a high-priority rule with a NULL action.

1.3.8 New ACL data type

A new ACL data type for CSIG signals is defined.

Table 1-11 - ACL CSIG Signal Data Type

typedef sai_csig_signal_type_t	Description
SAI_CSIG_SIGNAL_TYPE_ABW	Available Bandwidth in bytes per second
SAI_CSIG_SIGNAL_TYPE_ABWC	Normalized Available Bandwidth in percentage
SAI_CSIG_SIGNAL_TYPE_DELAY	Delay measure in nanoseconds
SAI_CSIG_SIGNAL_TYPE_NQD	Normalized queue depth
SAI_CSIG_SIGNAL_TYPE_COMPACT_EXP	Compact tag experimental
SAI_CSIG_SIGNAL_TYPE_WIDE_EXP	Wide tag experimental

1.3.9 Switch and Port Object Attributes

A new attribute is introduced at port level for the CSIG tag delete operation. Setting this flag instructs the switch pipeline to delete all the CSIG tags in the CSIG tagged packets egressing this port. Additionally, a metadata attribute is introduced to set the locator metadata for the CSIG tagged packets egressing on the port.

A switch attribute specifies the number of bandwidth timers a given implementation supports.

Table 1-12 - Switch and Port Attributes

Attribute	Type	Flags	Default	Description
SAI_PORT_ATTR_INT_DELETE	bool	CREATE_AND_SET	FALSE	Enable CSIG tag deletion for all the CSIG tag traffic egressing the port
SAI_PORT_ATTR_INT_METADATA	sai_uint16_t	CREATE_AND_SET	0	Locator metadata for all the traffic egressing the port
SAI_SWITCH_ATTR_MAX_CSIG_TIMERS	sai_uint8_t	READ_ONLY	disabled	Maximum number of timers supported for CSIG bandwidth and normalized bandwidth signal

1.3.10 CSIG Stats

All the CSIG statistics are managed using ACL counters. There are no new CSIG specific counters introduced in the SAI.

A CSIG tag with invalid signals SHOULD be counted using ACL rules, where a catch all rules can be installed for all unknown signal types along with a counter.

1.3.11 Capability Query

The capability query is a key component of SAI. A Network Operating System (NOS) first utilizes the capability query APIs to determine the capabilities of a given implementation. Based on this learned capability, the NOS then invokes SAI APIs to either configure the device or retrieve its operational state.

No new query APIs are introduced by this specification. The existing APIs are used to determine support for CSIG compact or wide tags, various signal types, the **csig.d** bit, and strip-and-forward functionalities.

1.4 Normative References

- [1] <https://github.com/opencomputeproject/SAI/pull/2248>
- [2] <https://groups.ultraethernet.org/wg/UE-TR/document/3586>