**IEEE P802.1CBdb**
(individual contribution)
**March 13, 2019**

# Frame Replication and Elimination for Reliability — Amendment:

# Extended Stream identification functions

**Abstract:** This standard specifies procedures, managed objects and protocols for bridges and end systems that provide identification and replication of packets for redundant transmission, identification of duplicate packets, and elimination of duplicate packets. It is not concerned with the creation of the multiple paths over which the duplicates are transmitted.
**Keywords:** TSN, Time-Sensitive Networking, Redundancy, Bridging, Bridges, Frame Replication, Frame Elimination, Bridged Local Area Networks, IEEE 802®, IEEE 802.1Q™, IEEE 802.1CB™, local area networks (LANs), MAC Bridges, Virtual Bridged Local Area Networks (virtual LANs).

# IEEE Draft Standard for Local and metropolitan area networks — Frame Replication and Elimination for Reliability — Amendment: Extended Stream Identification Functions

## 1. Overview

### 1.1 Scope

This amendment specifies procedures and managed objects that add new stream identification functions. Additionally this amendment addresses errors and clarifications.

### 1.2 Rationale

Stream identification is required by an increasing number of traffic management mechanisms implemented in Layer 2: ingress policing, traffic scheduling, congestion management, mapping to traffic classes, that make Ethernet networks suitable for a growing number of applications. Current stream identification methods defined in IEEE Std 802.1CB are insufficient for some of these applications.

### 1.6 Introduction

This amendment defines an additional passive stream identification function, which input parameters are not limited to the addressing parameters provided by the EISS indication primitive (destination_address, source_address and vlan_identifier) and, or, to some specific upper layer protocol (e.g. IP) information. The Extended Stream identification function is based on a mask-and-match scheme, where a set of masks is first applied to a subset of the parameters passed by the EISS primitive: destination_address, source_address, vlan_identifier and mac_service_data_unit. The resulting masked information fields are then compared against a set of values, one for each mask, that identify a particular Stream. Masks and match values are defined by managed objects.

<< Editor's Note: TBC >>

## 2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies. Non-normative references (i.e., that provide additional information not required for the application of this document) are given in Annex D.

IEEE Std 802®, IEEE Standard for Local and metropolitan area networks: Overview and Architecture.[1, 2]

IEEE Std 802.1AC™, IEEE Standard for Local and metropolitan area networks—Media Access Control (MAC) Service Definition.

IEEE Std 802.1Q™, IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks.

<< Editor's Note: other normative references will be added, as appropriate >>

[1]The IEEE standards or products referred to in Clause 2 are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.

[2]IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org).

# 3. Definitions

<< Editor's Note: Definitions will be added, as appropriate >>

**Mask:** aaabbb

# 4. Acronyms and abbreviations

<< Editor's Note: to be added to the existing list >>

| | |
|---|---|
| XXX | Text |
| YYYY | Text |

# 5. Conformance

This clause specifies the mandatory and optional capabilities provided by conformant implementations of this standard.

## 5.5 Stream identification component optional behaviors

***Insert new item d) as shown and re-number subsequent items:***

a) The items in 5.3 and 5.4 on more than one port;
b) The items in 5.3 and 5.4 for some number of Compound Streams greater than 1;
c) An IP Stream identification function (6.7); ~~and/or~~
d) An Extended Stream identification function (crossref); and/or
e) Additional types of Stream identification functions.

## 5.8 Talker end system optional behaviors

***Insert new item d) as shown and re-number subsequent items:***

a) The items in 5.8 and 5.7 on more than one port;
b) The items in 5.8 and 5.7 for some number of Compound Streams greater than 1;
c) An IP Stream identification function (6.7);
d) An Extended Stream identification function (crossref);
e) Additional types of Stream identification functions;
f) The HSR sequence tag (7.9);
g) The PRP sequence trailer (7.10); and/or
h) Additional types of Sequence encode/decode functions.

## 5.11 Listener end system optional behaviors

***Insert new item d) as shown and re-number subsequent items:***

a) The items in 5.11 and 5.10 on more than one port;
b) The items in 5.11 and 5.10 for some number of Compound Streams greater than 1;
c) An IP Stream identification function (6.7);
d) An Extended Stream identification function (crossref);
e) Additional types of Stream identification functions;
f) The HSR sequence tag (7.9);
g) The PRP sequence trailer (7.10);
h) Additional types of Sequence encode/decode functions; and/or
i) At least two instances of Individual recovery functions (7.5), each using the VectorRecoveryAlgorithm (7.4.3.4).

## 5.13 Relay system recommended behaviors

***Insert new item c) as shown:***

a) Active Destination MAC and VLAN Stream identification functions (6.6) for encoding and decoding packets; and
b) IP Stream identification functions (6.7); ~~or~~

c) Extended Stream identification function (crossref) for identifying packets.

NOTE—IP Stream identification enables a relay system to proxy for a FRER-unaware end system.

# 6. Stream identification

***Change the text of the first paragraph as shown:***

Clause 7 of IEEE Std 802.1AC describes the IEEE 802.1 layering model, that Frame Replication and Elimination for Reliability (FRER) follows. Stream identification utilizes a single Service Access Point (SAP) to a connectionless packet service offered by the layer below it [e.g., the Intermediate Sublayer Service (ISS) of Clause 11 of IEEE Std 802.1AC, or the Enhanced Internal Sublayer Service (EISS) of Clause 6.8 of IEEE Std], and offers an array of SAPs to the layers above it, corresponding to different Streams. The Stream identification model is illustrated in Figure 6-1.

***Change the text of the note as shown:***

NOTE—In principle, any number of different methods for identifying and encoding Streams can be defined. Several required methods are specified in the following subclauses (6.4, 6.5, 6.6, 6.9, crossref).

***Change the text of item c) and Table 6-1 as shown:***

c) ~~Four~~Five specific IP Stream identifications are described: Null Stream identification (6.4), Source MAC and VLAN Stream identification (6.5), Active Destination MAC and VLAN Stream identification (6.6), ~~and~~ IP Stream identification (6.9), and Extended Stream identification (crossref).

.

**Table 6-1—IP Stream identifications**

| IP Stream identification | Active/passive | Examines | Overwrites | Reference |
|---|---|---|---|---|
| Null Stream identification | Passive | destination_address, vlan_identifier | None | 6.4, 9.1.2 |
| Source MAC and VLAN Stream identification | Passive | source_address, vlan_identifier | None | 6.5, 9.1.6 |
| Active Destination MAC and VLAN Stream identification | Active | destination_address, vlan_identifier | destination_address, vlan_identifier, priority | 6.6, 9.1.4 |
| IP Stream identification | Passive | destination_address, vlan_identifier, IP source address, IP destination address, DSCP, IP next protocol, source port, destination port | None | 6.9, 9.1.5 |
| Extended Stream identification function | Passive | destination_address, source_address, vlan_identifier, mac_service_data_unit | None | 6.8, 9.1.6 |

<< Editor's Note: cross-references to be updated >>

## 6.1 Stream service subparameters

***Change the text of the first paragraph as shown:***

The ISS defined in IEEE Std 802.1AC—and the EISS defined in IEEE Std 802.1Q includes a connection_identifier parameter that is of local significance (to a system) only. The parameter is not carried across the underlying service. Stream identification makes use of this parameter to carry parametrized information. Stream identification has need for more than one subparameter, but an implementor can create mathematical algorithms to combine those subparameters (and/or other subparameters for other layers) into a single connection_identifier parameter, especially since the connection_identifier's values are undefined outside the system implementing them. In this document, parameters that are assumed to be encoded in the connection_identifier are deemed *subparameters*.

***Insert new clause 6.8 as shown:***

## 6.8 Extended Stream identification

The Extended Stream identification is a passive Stream identification that operates at the frame level. It can be defined using the Enhanced Internal Sublayer Service (EISS) described in 6.9 of IEEE Std 802.1Q-2018, in which case it is enhanced with the extra stream_handle subparameter of the connection_identifier, specified in 6.1 of the present standard.

It discards the stream_handle subparameter passed down the stack. It generates a stream_handle subparameter on frames passed up the stack based on information fields extracted from the frame's destination MAC address, source MAC address, VLAN ID and MAC service data unit. It does not change any of a packet's parameters. It is suitable for applications in which Streams are defined by a set of parameters derived from those provided through the EISS SAP. This parameter set includes a bit field of the destination MAC address, a bit field of the source MAC address, a bit field of the VLAN identifier and a set of bit fields of the MAC service data unit. In order to instantiate the Extended Stream identification function, the tsnStreamIdIdentificationType managed object (9.1.1.6) is encoded using the OUI (00-80-C2) and the type values as shown in Table 9-1.

Extended Stream identification can be coupled, for example, with Active Destination MAC and VLAN Stream identification (6.6) to assign a particular {MAC address, VLAN, priority} triplet to packets belonging to a particular unicast application Stream identified by the combination of the values of particular fields in the MAC service data unit, as shown in Figure 8-1, Port A, where Extended Stream identification would be in the box labeled "Passive Upper Extended Stream identification (6.8)." The managed objects for Extended Stream identification are described in (crossref 9.1.6).

NOTE—The drop_eligible parameter is also present, along with the VLAN identifier and priority, in an IEEE 802.1Q VLAN tag. FRER does not affect the use of this parameter. It passes through Extended Stream identification unchanged, and defaults to False when not present.

# 8. Frame Replication and Elimination for Reliability in Bridges

***Change paragraph of clause 8.2 as shown:***

## 8.2 FRER C-component input transformations

The Input transformations, marked with white boxed with boldface type in Figure 8-1, enable a Bridge to proxy for a non-FRER-capable end system. The expanded input port identifies packets belonging to a Stream (e.g., using IP Stream identification, 6.7, or Extended Stream identification, crossref 6.8), serializes the packets with a Sequence generation function (7.4.1), encodes the sequence number with an R-TAG (7.8), and then gives the packets belonging to this Stream a {vlan_identifier, destination_mac_address} pair that is unique, at least inside this Bridge, using Active Destination MAC and VLAN Stream identification (6.6). The IEEE 802.1Q Forwarding Process, enhanced with the Individual recovery function (7.5) and Sequence recovery function (7.4.2), then forwards the frame.

# 9. Stream Identification Management

## 9.1 Stream identity table

### 9.1.1 tsnStreamIdEntry

***Insert new row in Table 9-1 as shown:***

**Table 9-1—Stream identification types**

| OUI/CID | Type number | Stream identification function | Controlling parameters |
|---|---|---|---|
| 00-80-C2 | 0 | Reserved | — |
| 00-80-C2 | 1 | Null Stream identification (6.4) | 9.1.2 |
| 00-80-C2 | 2 | Source MAC and VLAN Stream identification (6.5) | 9.1.6 |
| 00-80-C2 | 3 | Active Destination MAC and VLAN Stream identification (6.6) | 9.1.4 |
| 00-80-C2 | 4 | IP Stream identification (6.7) | 9.1.5 |
| 00-80-C2 | 5 | Extended Stream identification (crossref 6.8) | (crossref 9.1.6) |
| 00-80-C2 | 5–255 | Reserved | — |
| other | — | Defined by entity owning the OUI or CID | — |

***Insert Clause 9.1.6 as shown:***

### 9.1.6 Managed objects for Extended Stream identification

When instantiating an instance of the Extended Stream identification function (crossref 6.8) for a particular input Stream, the managed objects in the following subclauses serve as the tsnStreamIdParameters managed object (9.1.1.2).

#### 9.1.6.1 tsnCpeEsIdDestMacMask

Specifies a 48-bit mask to be applied to the destination_address parameter passed by the EISS indication primitive to the Extended Stream identification function. A bit set to ’1’ in tsnCpeEsIdDestMacMask indicates that the bit with the same position in the destination_address parameter must match the bit with the same position in tsnCpeEsIdDestMacMatch (crossref 9.1.6.2). A bit set to ’0’ in tsnCpeEsIdDestMacMask indicates that the bit with the same position in the destination_address parameter is ignored. An all-0 tsnCpeEsIdDestMacMask indicates that the destination_address parameter is ignored.

#### 9.1.6.2 tsnCpeEsIdDestMacMatch

Specifies the 48-bit value that the destination_address parameter, passed by the EISS indication primitive to the Extended Stream identification function, must match according to the masking rules defined in (crossref 9.1.6.1).

#### 9.1.6.3 tsnCpeEsIdSrcMacMask

Specifies a 48-bit mask to be applied to the source_address parameter passed by the EISS indication primitive to the Extended Stream identification function. A bit set to ’1’ in tsnCpeEsIdSrcMacMask indicates that the bit with the same position in the source_address parameter must match the bit with the same position in tsnCpeEsIdSrcMacMatch (crossref 9.1.6.4). A bit set to ’0’ in tsnCpeEsIdSrcMacMask indicates that the bit with the same position in the source_address parameter is ignored. An all-0 tsnCpeEsIdSrcMacMask indicates that the source_address parameter is ignored.

#### 9.1.6.4 tsnCpeEsIdSrcMacMatch

Specifies the 48-bit value that the source_address parameter, passed by the EISS indication primitive to the Extended Stream identification function, must match according to the masking rules defined in (crossref 9.1.6.3).

#### 9.1.6.5 tsnCpeEsIdTagged

An enumerated value indicating whether a frame in an EISS indication primitive to the Extended Stream identification function is permitted to have a VLAN tag. It can take the following values:

1) **tagged:** A frame must have a VLAN tag to be recognized as belonging to the Stream.
2) **priority:** A frame must be untagged, or have a VLAN tag with a VLAN ID = 0 to be recognized as belonging to the Stream.
3) **all:** A frame is recognized as belonging to the Stream whether tagged or not.

<< Editor’s Note: this object provides a work-around to determine if a frame is VLAN-tagged. >>

#### 9.1.6.6 tsnCpeEsIdVlanIdMask

Specifies a 12-bit mask to be applied to the vlan_identifier parameter passed by the EISS indication primitive to the Extended Stream identification function. A bit set to ’1’ in tsnCpeEsIdVlanIdMask indicates that the bit with the same position in the vlan_identifier parameter must match the bit with the same position in tsnCpeEsIdVlanIdMatch (crossref 9.1.6.7). A bit set to ’0’ in tsnCpeEsIdVlanIdMask indicates that the bit with the same position in the vlan_identifier parameter is ignored. An all-0 tsnCpeEsIdVlanIdMask indicates that the vlan_identifier parameter is ignored.

#### 9.1.6.7 tsnCpeEsIdVlanIdMatch

Specifies the 12-bit value that the vlan_identifier parameter, passed by the EISS indication primitive to the Extended Stream identification function, must match according to the masking rules defined in (crossref 9.1.6.6).

#### 9.1.6.8 tsnCpeEsIdMsduFieldNb

Specifies the number of bit fields in the mac_service_data_unit parameter of the EISS indication primitive to the Extended Stream identification function to be used to identify the Stream. A value of 0 indicates that the mac_service_data_unit parameter is not used by the Extended Stream identification function. tsnCpeEsIdMsduFieldNb has a maximum value of XXX. If tsnCpeEsIdMsduFieldNb has a value N greater than 0, then N bit fields shall be defined using N mask definitions (tsnCpeEsIdMsduField1, ..., tsnCpeEsIdMsduFieldN (crossref 9.1.6.9)), and N matching values shall be defined by tsnCpeEsIdMsduFieldValue1, ..., tsnCpeEsIdMsduFieldValueN (crossref 9.1.6.10)

<< Editor’s Note: XXX to be fixed. >>

### 9.1.6.9 tsnCpeEsIdMsduFieldN

Specifies a series of consecutive bits of the mac_service_data_unit parameter passed by the EISS indication primitive to the Extended Stream identification function. tsnCpeEsIdMsduFieldN is defined using two other objects indicating the location (tsnCpeEsIdMsduFieldOffsetN, (crossref 9.1.6.9.1)) and length (tsnCpeEsIdMsduFieldLengthN, (crossref 9.1.6.9.2) of the bit field.

#### 9.1.6.9.1 tsnCpeEsIdMsduFieldOffsetN

Specifies the offset, expressed in bits and relative to the first bit of the mac_service_data_unit parameter, of the first bit of tsnCpeEsIdMsduFieldN. A value of i indicates that the first bit of tsnCpeEsIdMsduFieldN is the $(i+1)^{th}$ bit of the mac_service_data_unit parameter. tsnCpeEsIdMsduFieldOffsetN has a value comprised between 0 and XXX.

<< Editor's Note: XXX has to be fixed >>

#### 9.1.6.9.2 tsnCpeEsIdMsduFieldLengthN

Specifies the length, expressed in bits, of tsnCpeEsIdMsduFieldN. tsnCpeEsIdMsduFieldLengthN has a value comprised between 1 and XXX.

<< Editor's Note: XXX has to be fixed >>

### 9.1.6.10 tsnCpeEsIdMsduFieldValueN

Specifies the value of tsnCpeEsIdMsduFieldN, as defined in (crossref 9.1.6.9), that participates to the Stream identification by the Extended Stream identification function.

# Annex A

## (normative)

# Protocol Implementation Conformance Statement (PICS) proforma

## A.2 PICS proforma for Frame Replication and Elimination for Reliability

### A.2.2 Stream identification component

***Insert item IS5 and re-number subsequent entries in Table as shown:***

| Item | Feature | Subclause | Value/Comment | Status | Support |
|---|---|---|---|---|---|
| IS1 | Can the system identify frames using the Null Stream identification function? | 5.3:b, 6.4 | | IS: M | Yes [ ] |
| IS2 | Does the system implement the required managed objects of Clause 9? | 5.3:c, 9 | | IS: M | Yes [ ] |
| IS3 | Can the system encode frames using the Active Destination MAC and VLAN Stream identification? | 5.4:a, 6.6 | | IS: O | Yes [ ]<br>No [ ]<br>—[1] |
| IS4 | Can the system identify packets using the IP Stream identification? | 5.5:c, 6.7 | | IS: O | Yes [ ]<br>No [ ] |
| IS5 | Can the system identify packets using the Extended Stream identification [crossref]? | 5.5:d, 6.8 | | IS: O | Yes [ ]<br>No [ ] |
| IS6 | For what additional Stream decodings can the system be configured? | 5.5:e | | IS: O | — |
| IS7 | Explain the limits on which ports the above features can be configured. | 5.5:a | | IS: O | — |
| IS8 | Explain the limits on the number of Streams for which the above features can be configured. | 5.5:b | | IS: O | — |

[1]If "No," supply a reason why.

### A.2.3 Talker end system

***Insert item TE16 and re-number subsequent entries in Table as shown:***

| Item | Feature | Subclause | Value/Comment | Status | Support |
|---|---|---|---|---|---|
| TE9 | Can the system identify frames using the Null Stream identification function? | 5.6:b, 6.4 | | TE: M | Yes [ ] |
| TE10 | Can the system be configured with a Sequence generation function? | 5.6:c, 7.4.1 | | TE: M | Yes [ ] |
| TE11 | Can the system be configured with a Sequence encode/decode function? | 5.6:d, 7.8 | | TE: M | Yes [ ] |
| TE12 | Does the system implement the managed objects of Clause 9 and Clause 10 (10.7 not required)? | 5.6:e, 9, 10 | | TE: M | Yes [ ] |
| TE13 | Can the system encode frames using the Active Destination MAC and VLAN Stream identification? | 5.7:a, 6.6 | | TE: O | Yes [ ]<br>No [ ]<br>__[1] |
| TE14 | Can the system be configured with a Stream splitting function? | 5.7:b, 7.7 | | TE: M | Yes [ ]<br>No [ ]<br>__[a] |
| TE15 | Can the system identify packets using the IP Stream identification? | 5.8:c, 6.7 | | TE: O | Yes [ ]<br>No [ ] |
| TE16 | Can the system identify packets using the Extended Stream identification [crossref]? | 5.8:d, 6.8 | | TE: O | Yes [ ]<br>No [ ] |
| TE17 | For what additional Stream decodings can the system be configured? | 5.8:e | | TE: O | — |
| TE18 | Can the system encode frames using HSR sequence tag? | 5.8:f, 7.9 | | TE: O | Yes [ ]<br>No [ ] |
| TE19 | Can the system encode frames using PRP sequence trailer? | 5.8:g, 7.10 | | TE: O | Yes [ ]<br>No [ ] |
| TE20 | For what additional Sequence encode/decode functions can the system be configured? | 5.8:h | | TE: O | — |
| TE21 | Explain the limits on which ports the above features can be configured. | 5.8:a | | TE: O | — |
| TE22 | Explain the limits on the number of Streams for which the above features can be configured. | 5.8:b | | TE: O | — |

[1]If "No," supply a reason why.

### A.2.4 Listener end system

***Insert item LE11 and re-number subsequent entries in Table as shown:***

| Item | Feature | Subclause | Value/Comment | Status | Support |
|---|---|---|---|---|---|
| LE1 | Can the system identify frames using the Null Stream identification? | 5.9:b, 6.4 | | LE: M | Yes [ ] |
| LE2 | Can the system be configured with at least two Individual recovery functions? | 5.9:c, 7.5 | | LE: M | Yes [ ] |
| LE3 | Can the system be configured with at least one Sequence recovery function using the MatchRecoveryAlgorithm? | 5.9:c, 7.4.2, 7.4.3.5 | | LE: M | Yes [ ] |
| LE4 | Does the system support the Sequence recovery function using the VectorRecoveryAlgorithm with a value of frerSeqRcvyHistoryLength ≥ 2? | 5.9:c, 7.4.2, 7.4.3.4 | | LE: M | Yes [ ] |
| LE5 | Can the system be configured with at least two Individual recovery functions using the MatchRecoveryAlgorithm? | 5.9:d, 7.5, 7.4.3.5 | | LE: M | Yes [ ] |
| LE6 | Can the system be configured with a Sequence decoding function? | 5.9:e, 7.8 | | LE: M | Yes [ ] |
| LE7 | Does the system implement the managed objects of Clause 9 and Clause 10 (10.7 not required)? | 5.9:f, 9, 10 | | LE: M | Yes [ ] |
| LE8 | Does the Base recovery function process a frame before its FCS has been verified? | 7.4.3 | | LE: M | No [ ] |
| LE9 | Can the system decode frames using the Active Destination MAC and VLAN Stream identification? | 5.10:a, 6.6 | | LE: O | Yes [ ]<br>No [ ]<br>—[1] |
| LE10 | Can the system decode packets using the IP Stream identification? | 5.11:c, 6.7 | | LE: O | Yes [ ]<br>No [ ] |
| LE11 | Can the system identify packets using the Extended Stream identification [crossref]? | 5.11:d, 6.8 | | LE: O | Yes [ ]<br>No [ ] |
| LE12 | For what additional Stream decodings can the system be configured? | 5.11:e | | LE: O | — |
| LE13 | Can the system decode frames using HSR sequence tag? | 5.11:f, 7.9 | | LE: O | Yes [ ]<br>No [ ] |
| LE14 | Can the system decode frames using PRP sequence trailer? | 5.11:g, 7.10 | | LE: O | Yes [ ]<br>No [ ] |

| Item | Feature | Subclause | Value/Comment | Status | Support |
|---|---|---|---|---|---|
| LE15 | For what additional Sequence decodings can the system be configured? | 5.11:h | | LE: O | — |
| LE16 | Can the system be configured with at least two Individual recovery functions using the VectorRecoveryAlgorithm? | 5.11:i, 7.5, 7.4.3.4 | | LE: O | Yes [ ] |
| LE17 | Explain the limits on which ports the above features can be configured. | 5.11:a | | LE: O | — |
| LE18 | Explain the limits on the number of Streams for which the above features can be configured. | 5.11:b | | LE: O | — |

[1]If "No," supply a reason why.

## A.2.5 Relay system

***Insert item RS12 and re-number subsequent entries in Table as shown:***

| Item | Feature | Subclause | Value/Comment | Status | Support |
|---|---|---|---|---|---|
| RS1 | Can the system identify frames using the Null Stream identification function? | 5.12:b, 6.4 | | RS: M | Yes [ ] |
| RS2 | Can the system be configured with a Sequence generation function? | 5.12:c, 7.4.1 | | RS: M | Yes [ ] |
| RS3 | Can the system be configured with at least two Individual recovery functions? | 5.12:e, 7.5 | | RS: M | Yes [ ] |
| RS4 | Can the system be configured with at least one Sequence recovery function using the MatchRecoveryAlgorithm? | 5.12:e, 7.4.2, 7.4.3.5 | | RS: M | Yes [ ] |
| RS5 | Does the system support the Sequence recovery function using the VectorRecoveryAlgorithm with a value of frerSeqRcvyHistoryLength ≥ 2? | 5.12:e, 7.4.2, 7.4.3.4 | | RS: M | Yes [ ] |
| RS6 | Can the system be configured with at least two Individual recovery functions using the MatchRecoveryAlgorithm? | 5.12:f, 7.5, 7.4.3.5 | | RS: M | Yes [ ] |
| RS7 | Can the system be configured with a Sequence encode/decode function? | 5.12:d, 7.8 | | RS: M | Yes [ ] |

| Item | Feature | Subclause | Value/Comment | Status | Support |
|---|---|---|---|---|---|
| RS8 | Does the system implement the managed objects of Clause 9 and Clause 10 (including 10.7)? | 5.12:g, 9, 10 | | RS: M | Yes [ ] |
| RS9 | Does the Base recovery function process a frame before its FCS has been verified? | 7.4.3 | | RS: M | No [ ] |
| RS10 | Can the system encode/decode frames using the Active Destination MAC and VLAN Stream identification? | 5.13:a, 6.6 | | RS: O | Yes [ ]<br>No [ ]<br>—[1] |
| RS11 | Can the system identify packets using the IP Stream identification? | 5.13:b, 6.7 | | RS: O | Yes [ ]<br>No [ ]<br>—[a] |
| RS12 | Can the system identify packets using the Extended Stream identification [crossref]? | 5.13:c, 6.8 | | RS: O | Yes [ ]<br>No [ ]<br>—[a] |
| RS13 | For what additional Stream identification functions can the system be configured? | 5.14:c | | RS: O | — |
| RS14 | Can the Stream splitting function be configured on the system? | 5.14:d, 7.7 | | RS: O | Yes [ ]<br>No [ ] |
| RS15 | Can the system encode/decode frames using HSR sequence tag? | 5.14:e, 7.9 | | RS: O | Yes [ ]<br>No [ ] |
| RS16 | Can the system encode/decode frames using PRP sequence trailer? | 5.14:f, 7.10 | | RS: O | Yes [ ]<br>No [ ] |
| RS17 | For what additional Sequence encode/decode functions can the system be configured? | 5.14:g | | RS: O | — |
| RS18 | Can the system be configured with at least two Individual recovery functions using the VectorRecoveryAlgorithm? | 5.14:i, 7.5, 7.4.3.4 | | RS: O | Yes [ ]<br>No [ ] |
| RS19 | Can the system be configured for Autoconfiguration via the Managed objects for autoconfiguration? | 5.14:j, 7.11, 10.7 | | RS: O | Yes [ ]<br>No [ ] |
| RS20 | Explain the limits on which ports the above features can be configured. | 5.14:a | | RS: O | — |
| RS21 | Explain the limits on the number of Streams for which the above features can be configured. | 5.14:b | | RS: O | — |
| RS22 | Explain the limits on whether the above features can be configured at in-facing or out-facing positions. | 5.14:h | | RS: O | — |

[1]If “No,” supply a reason why.

1

# Annex C

## (informative)

# Frame Replication and Elimination for Reliability in systems

## C.2 Example 2: Various stack positions

### *Change 3rd paragraph of clause C.2, and Figure C-5 as shown:*

Figure C-5 illustrates relay system B in Figure C-4. As the packets enter from the left, from End System A, they pass first through a Stream identification function [IP Stream identification (6.7), or Extended Stream identification (crossref 6.8)], which identifies the Stream. The Stream Transfer Function delivers the packet with all TSN parameters, including the stream_handle subparameter, to the Sequence generation function (7.4.1, marked "Seq." in Figure C-4), which adds a sequence_number subparameter with a steadily-increasing integer sequence value (modulo the size of the packet field carrying the sequence_number). The sequence_number subparameter is encapsulated into the packet by the Sequence encode/decode function (7.6). A Stream identification function [this time, Active Destination MAC and VLAN Stream identification (6.6)] modifies the two packets' destination MAC addresses and VLANs for identification through the bridged network. Relay system B's forwarding function then outputs the two packets on two different ports. The external form of the packets are labeled differently, as indicated by the italic numbers *26* and *31* in Figure C-4.

.

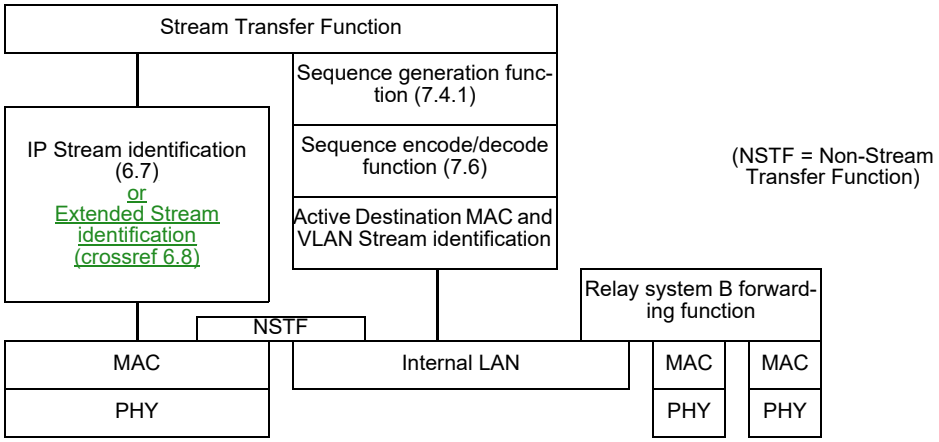


**Figure C-5—Protocol stack for relay system B, proxying for End System A, in Figure C-4**

## C.5 Example 5: Protocol interworking

### *Change 1st paragraph of clause C.5 as shown:*

Figure C-10 illustrates a simple protocol interworking function in one port of a relay system. In this example, two different encapsulation schemes **1** and **2** are used for the two legs of the Stream Transfer

Function, so that packets are transformed from using one encapsulation to using the other encapsulation as they pass through the port. No additional functions, e.g., a Sequence recovery function (7.4.2) are shown, although they would be perfectly admissible. If this were a port of a bridge attached to an end system, encapsulation **1** could be the Active Destination MAC and VLAN Stream identification (6.6), and encapsulation **2** could be the IP Stream identification (6.7) or the Extended Stream identification (crossref 6.8). The net result for the end system could be to convert a specific unicast IP Stream to use a specific multicast destination address and VLAN, in order to direct the packet through a specific path through the bridged network. Presumably, a similar interworking pair at the other end of the Stream would restore the packet to its original destination MAC address and VLAN.