802.1CB-2017/Cor1

Editor's update

Christophe Mangin

2024/7/16

MITSUBISHI ELECTRIC R&D CENTRE EUROPE

Content

- Maintenance items
  - Items related to 802.1CB-2017
    - Item #315
    - Item #344
  - Items related to 802.1CBcv-2021
    - Item #338
    - Item #364
  - Item related to 802.1CB-2017, partly resolved and not in PAR
    - Item #343
- Draft 0.0
  - overview
  - New editorial maintenance items
    - Ex.: note added by 802.1CBdb not numbered
    - What to do ?

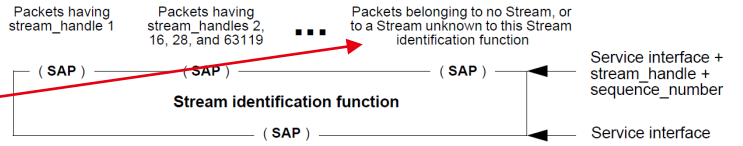
802.1CB-2017, item #315

- Clarification on stream_handle's « null-value »
  - Description
    - As discussed, the intention is that the stream_handle and sequence number are not provided if there is no match. The stream_handle is an internal value that is not exchanged externally so there is no need to define an explicit value for it. The sequence_number is sent externally and a value of 0 is valid. The proposal is to change the text to indicate the intention and avoid the use of 'null value'. Consider changing "null" to "no".
  - Proposed resolution
    - Define "null value" as the numerical value 0.
  - Proposed text (*inconsistency ?*)
    - New paragraph at the end of Cl. 6.2:

    "The null-value, i.e., the stream_handle assigned to packets that belong to no known Stream, is 0."

    - New note at the end of Cl. 9.1.1.1 tsnStreamIdHandle:

    "NOTE - It is also possible to assign a packet that was identified as belonging to a known Stream (i.e., a specific tsnStreamIdEntry) a stream_handle equal 0, i.e., the null-value."
  - Editor's thought
    - Has this change an impact on the NSTF/STF functions and Stream identification functions' upper SAPs ?



- Handling of R-TAG in IP Stream Identification
  - Description
    - The IP Stream Identification must be able to locate the IP Ethertype (0x0800/0x86DD) in the Frame in order to find Layer 3 (addresses) and Layer 4 (ports) matching parameters. If an R-Tag is present (Figure 8-3), this IP Ethertype is in a different position, than if not. The text of 6.7 and the managed objects (9.1.5) do not give any indication of how the IP Stream Identification should handle this case. How far back in the frame should the IP Stream Identification look for the IP Ethertype? Could there be other Tags present?.
  - Proposed resolution
    - Add some text or even a configuration item giving some indication where the IP Ethertype is to be looked for by IP Stream Identification if (no) other tags (specifically the R-Tag from this standard) than the C-VLAN are present.
  - Editor's proposal
    - New paragraph after the 2nd paragraph of Cl. 6.2:

"Practically, the IP Stream identification function requires to locate the beginning of the IP packet encapsulated in the frame, preceeded by an IP Ethertype field, e.g. 0x0800 or 0X88DD. The location of the IP Ethertype field within the frame is dependent on the combination of Tag(s) following the addresses in a frame. The method used to determine this location is beyond the scope of this standard".

- Inaccurate description of URN usage in published 802.1CBcv YANG clause
  - Description
    - IEEE Std P802.1CBcv Cl. 12.1, "YANG Framework", includes: "The YANG framework applies hierarchy in the following areas: 1) The uniform resource name (URN), as specified in IEEE Std 802d-2017. The structure of the URN is such that ieee is the root (i.e., name-space identifier), followed by the standard, then the working group developing the standard."
    - The description of the structure of the URN does not appear to match the URNs actually used, nor does is it agree with the specification in IEEE Std 802-d nor https://standards.ieee.org/wp-content/uploads/import/documents/tutorials/ieeeurn.pdf.
      Here are some examples of urn usage from published 802.1Q YANG modules: namespace urn:ieee:std:802.1Q:yang:ieee802-types; namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-types; namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge; namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-stream-filters-gates; This usage does agree with Clause 11 of IEEE Std 802-d.

  - Proposed text
    - In Cl. 12.1 :

    Remove the sentence "The structure of the URN is such that ieee is the root (i.e., name-space identifier), followed by the standard, then the working group developing the standard".

- Stream split table in IEEE 802.1CBcv-2021 FRER YANG
  - Description
    - IEEE 802.1CS-2017 defines the Stream split table in 10.6 as : “There is one Stream split table per system, with one frerSplitEntry (10.6.1) per Stream splitting function (7.7) per set of stream_handle values.”
    - In the FRER YANG module, the Stream split table is realized as stream-split list, which uses port and direction-out-facing as index. This definition does not reflect the definition of the Stream split table in 10.6 of IEEE 802.1CB-2017. With this definition, only a single Stream can be split at an end system port.
  - Proposed YANG change
    - Modified “stream-split-table” specification :

```
+--rw stream-split-table* [stream-split-index]
|  +--rw stream-split-index integer
|  +--rw port if:interface-ref
|  +--rw direction-out-facing dot1cb-sid-types:direction
|  +--rw input-id*  -&gt; /dot1cb-sid:stream-identity/handle
|  +--rw output-id* -&gt; /dot1cb-sid:stream-identity/handle
```

- Stream split table in IEEE 802.1CBcv-2021 FRER YANG
  - Proposed implementation
    - Cl. 12.2.2 Frame Replication and Elimination for Reliability model
      - Figure 12-5—FRER model

stream-split (10.6)		
if-ref	port	// r-w
boolean	direction-out-facing	// r-w
leaf-ref	* input-id	// r-w
leaf-ref	* output-id	// r-w

stream-split (10.6)		
uint32	index	// r-w
if-ref	port	// r-w
boolean	direction-out-facing	// r-w
leaf-ref	*input-id	// r-w
leaf-ref	*output-id	// r-w

    - Cl. 12.6.1.2 YANG data scheme definition for ieee802-dot1cb-frer YANG module
      - Figure 12-5—FRER model

```
+--rw stream-split* [port direction-out-facing]
|  +--rw port                    if:interface-ref
|  +--rw direction-out-facing    dot1cb-sid-types:direction
|  +--rw input-id*               -> /dot1cb-sid:stream-identity/handle
|  +--rw output-id*              -> /dot1cb-sid:stream-identity/handle
```

```
+--rw stream-split* [index]
|  +--rw index                   uint32
|  +--rw port                    if:interface-ref
|  +--rw direction-out-facing    dot1cb-sid-types:direction
|  +--rw input-id*               -> /dot1cb-sid:stream-identity/handle
|  +--rw output-id*              -> /dot1cb-sid:stream-identity/handle
```

- Stream split table in IEEE 802.1CBcv-2021 FRER YANG
  - Proposed implementation
    - Cl. 12.6.2.4 YANG data scheme definition for ieee802-dot1cb-frer YANG module

```
revision 2021-12-08 {
  description
    "Published as part of IEEE Std 802.1CBcv-2021. Initial version.";
  reference
    "IEEE Std 802.1CBcv-2021, Frame Replication and Elimination for
    Reliability - FRER YANG Data Model and Management Information Base
    Module.";
}
```

```
revision 2024-MM-DD {
  description
    "Published as part of IEEE Std 802.1CB-2017/Cor 1-2025.";
  reference
    "IEEE Std 802.1CB-2017/Cor 1-2025, Frame Replication and Elimination for
    Reliability - Corrigendum 1.";
}
revision 2021-12-08 {
  description
    "Published as part of IEEE Std 802.1CBcv-2021. Initial version.";
  reference
    "IEEE Std 802.1CBcv-2021, Frame Replication and Elimination for
    Reliability - FRER YANG Data Model and Management Information Base
    Module.";
}
```

MITSUBISHI ELECTRIC
Changes for the Better

- Stream split table in IEEE 802.1CBcv-2021 FRER YANG

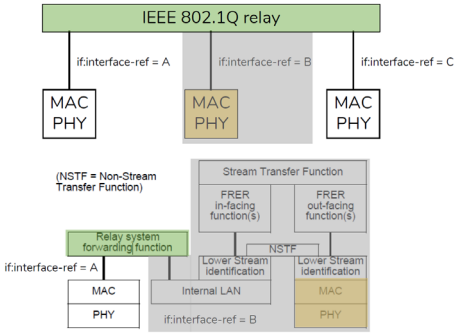
```
list stream-split {
  key "port direction-out-facing";
  description
    "There is one Stream split table per system, with one
    frerSplitEntry per Stream splitting function per set of
    stream_handle values. Each entry in the Stream split table
    specifies a port and direction on which an instance of the Stream
    splitting function is to be instantiated, and the list of
    stream_handles specifying its operation.";
  reference
    "10.6 of IEEE Std 802.1CB-2017";
  leaf port {
    type if:interface-ref;
    description
      "The port on which the system is to place an instance of the
      Stream splitting function performing the stream_handle
      translations specified by frerSplitInputIdList and
      frerSplitOutputIdList is to be placed.";
    reference
      "10.6.1.1 of IEEE Std 802.1CB-2017";
  }
  leaf direction-out-facing {
    type dot1cb-sid-types:direction;
    description
      "An object indicating whether the instance of the Stream
      splitting function performing the stream_handle translations
      specified by frerSplitInputIdList and frerSplitOutputIdList is
      to be placed on the out-facing (True) or in-facing (False) side
      of the port.";
    reference
      "10.6.1.2 of IEEE Std 802.1CB-2017";
  }
```

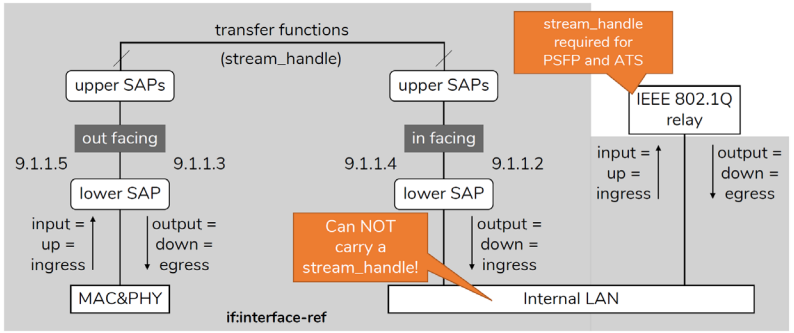
```
list stream-split {
  key "index";
  description
    "There is one Stream split table per system, with one
    frerSplitEntry per Stream splitting function per set of
    stream_handle values. Each entry in the Stream split table
    specifies a port and direction on which an instance of the Stream
    splitting function is to be instantiated, and the list of
    stream_handles specifying its operation.";
  reference
    "10.6 of IEEE Std 802.1CB-2017";
  leaf index {
    type uint32;
    description
      "Entry in the Stream split table.";
  }
```

- Handling of Stream_handle in Active Stream Identification
  - Initial description

    https://www.ieee802.org/1/files/public/docs2022/cb-turner-maint0343-1022-v01.pdf

    - All the passive stream identification functions (6.4, 6.5, 6.7, 6.8) have a statement along the lines of: “It discards the stream_handle subparameter passed down the stack”.
      For 6.6 Active Destination MAC and VLAN Stream identification such a statement is missing. It is unclear if it shall also discard the stream_handle on the down path (as suggested in 6.2: “The other SAP connects to the lower layers. This SAP can, but typically does not, include the stream_handle”) or not.





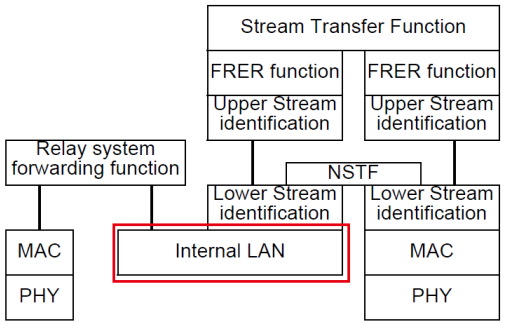
  - Several issues have been further pointed out

    https://www.ieee802.org/1/files/public/docs2023/CB-turner-StreamHandleMaintID0343-0623-v01.pdf

    - Stream_handle “loss” in an ingress port along the path towards the bridge forwarding functions
    - Introduction of “Lower” and “Upper” Stream identification functions.
    - Difference (inconsistency ?) between the layered model used to describe the location of Stream identification and FRER functions in a relay system and the arrangement of these functions in a bridge; migration of some forwarding/relay functions into the MAC.

- Stream_handle “loss” in an ingress port along the path towards the bridge forwarding functions
  - Description

    1. An in-facing Stream Identification Function will currently discard the stream_handle subparameter upon output towards the Internal LAN (6.4, 6.5, 6.7, 6.8).
    Even if no in-facing Stream Identification Function is configured, section 6.2 implies the stream_handle subparameter is not passed towards the Internal LAN upon output.
    This prevents the stream_handle subparameter to be available for the Per-stream classification and metering ([Q]:8.6.5.2).



  - Initial proposed resolution
    - Remove the sentence “It discards the stream_handle subparameter passed down the stack” in 6.4, 6.5, 6.7, 6.8 (passive Stream identification functions)
    - Add a new sub-clause (6.x) describing the Internal LAN and the Forwarding functions

“The Internal LAN models a fully transparent frame transfer function along with all (Enhanced) Internal Sublayer Service (E/ISS) parameters.”

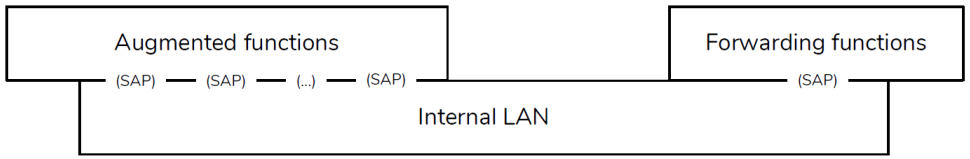


Figure 6-y—Internal LAN: array of upper SAPs

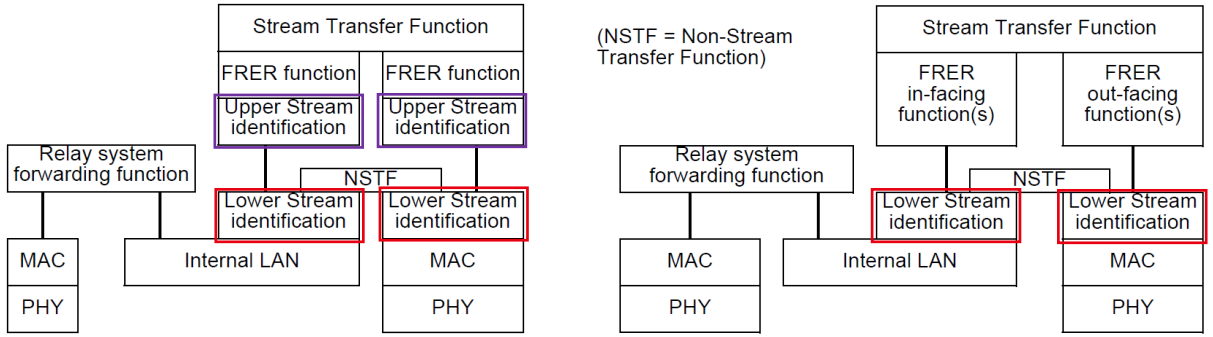
“The Forwarding functions entail the functionality of [Q] sections 8.6.1 through 8.6.5, but not [Q] Clause 6, [Q] section 8.5, nor [Q] sections 8.6.6 through 8.6.8, as those are modelled in the MAC (see Figure 6-6).”

- Stream_handle “loss” in an ingress port along the path towards the bridge forwarding functions
  - Editor’s proposal
    - Keep consistency with existing text of 6.3 already introducing the (N)STF
      - No additional sub-clause for the definition of the Internal LAN
    - Include the following text in the 2nd paragraph of 6.3 after « … in order to enable an FRER function ».

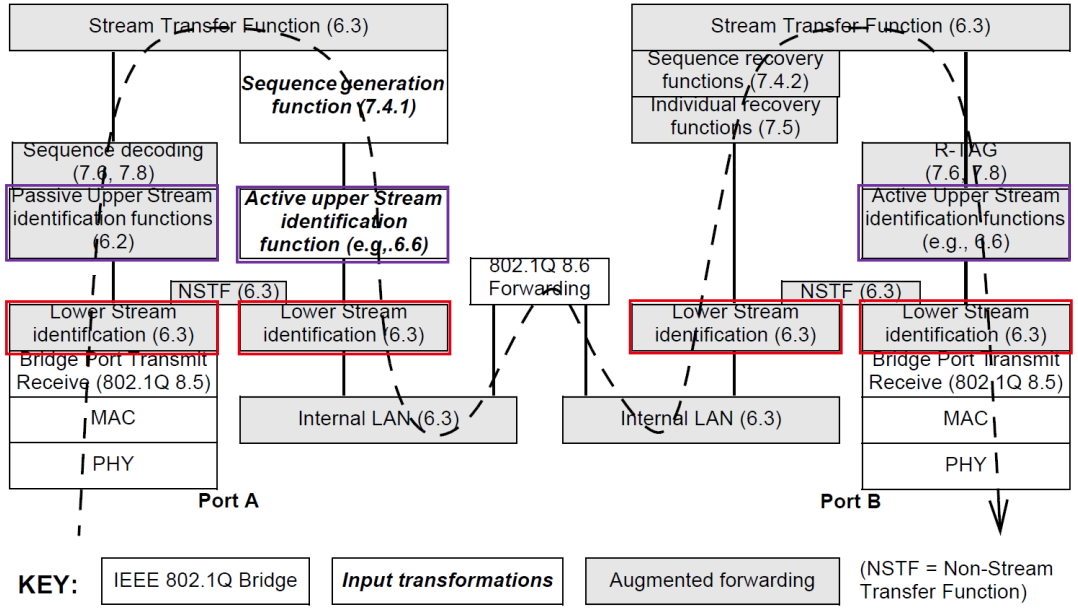
      « The connectivity between the Relay system’s forwarding and the Stream identification functions is modelled by an Internal LAN that provides transparent connectivity between ISS, or EISS, SAPs supporting the Stream service subparameters (6.1). »
    - Remove the sentence “It discards the stream_handle subparameter passed down the stack” in 6.4, 6.5, 6.7, 6.8
    - Any other changes required ?

- Introduction of “Lower” and “Upper” Stream identification functions
  - Description

3. Figures 6-5, 6-6, and 8-1 show a Lower and an Upper Stream identification function: “The Lower Stream identification functions separate FRER packets from non-FRER packets; the latter are relayed across the NSTF as if the FRER capabilities were not present. The Upper Stream identification functions identify the FRER packets’ Streams …” (6.3), while Figure 6-4 shows a single identification block with upper layer SAPs for the different stream_handles and the NSTF (“Packets belonging to no Stream, or to a Stream unknown to this Stream identification function”).
The task to “**separate FRER packets from non-FRER packets**” (6.3) is impossible to achieve, as only the match to a Stream Identification function constitutes the Frame to have to be processed by further FRER functionality up the stack.



b. Relay system with separated FRER capability, e.g., packet counting, on an output port.



- Introduction of “Lower” and “Upper” Stream identification functions
  - Initial proposed resolution
    - Change text in 6.3

The “Lower/Upper Stream identification function” terminology should be removed from 6.3 and Figure 6-3 deleted.

~~The Lower Stream identification functions separate FRER packets from non-FRER packets; the latter are relayed across the NSTF as if the FRER capabilities were not present~~. The ~~Upper~~Stream identification functions identify the FRER packets’ Streams so that the other FRER functions can perform their tasks and separate non-FRER packets, which are relayed across the NSTF.

    - Changes to figures in 6.2

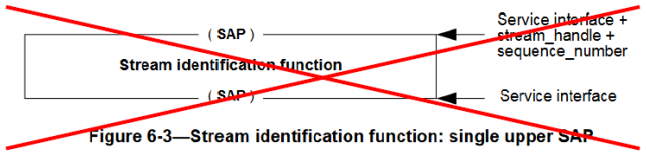


Figure 6-3—Stream identification function: single upper SAP

Delete Figure 6-3

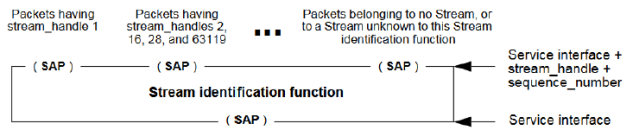


Figure 6-4—Stream identification function: array of upper SAPs

Modify Figure 6-4

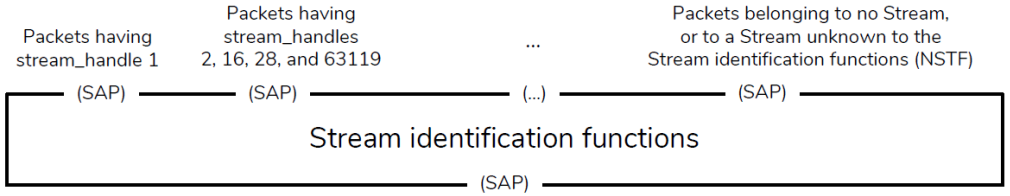
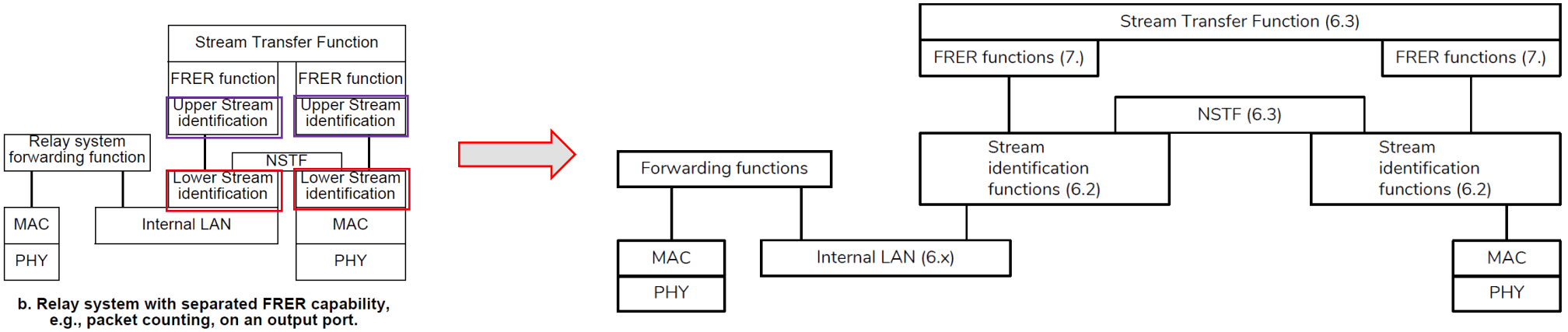


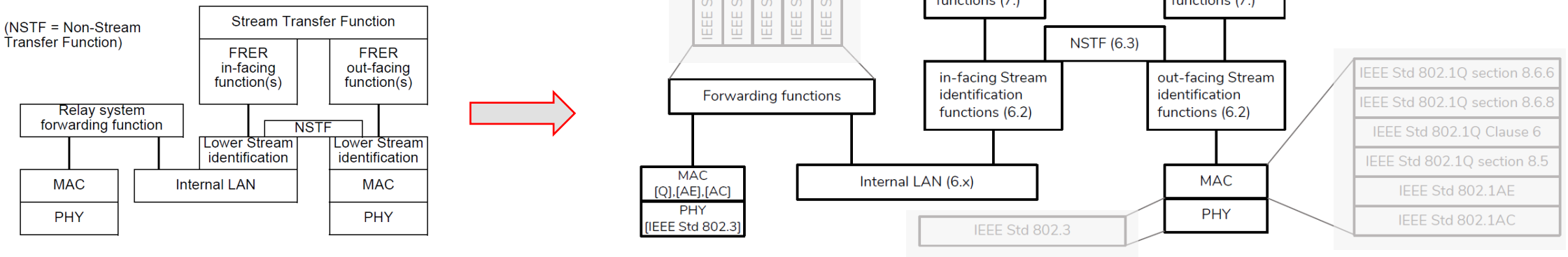
Figure 6-4—Stream identification functions

- Introduction of “Lower” and “Upper” Stream identification functions
  - Initial proposed resolution
    - Changes to figures in 6.5b

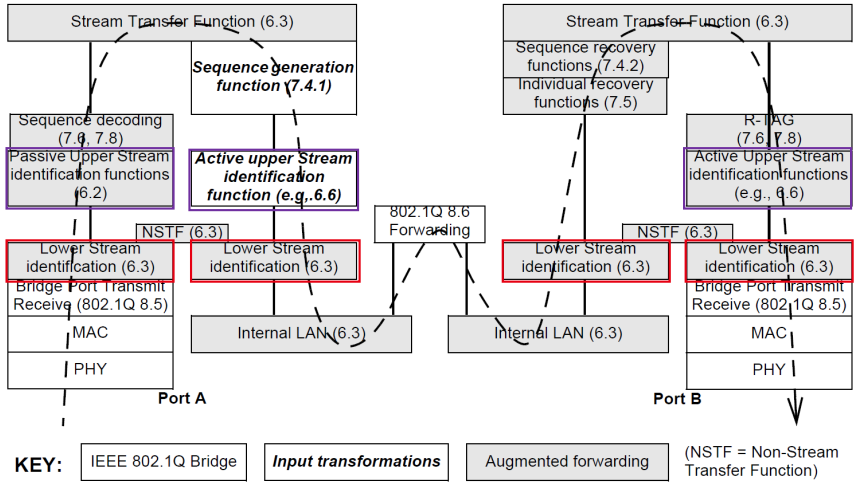


b. Relay system with separated FRER capability, e.g., packet counting, on an output port.

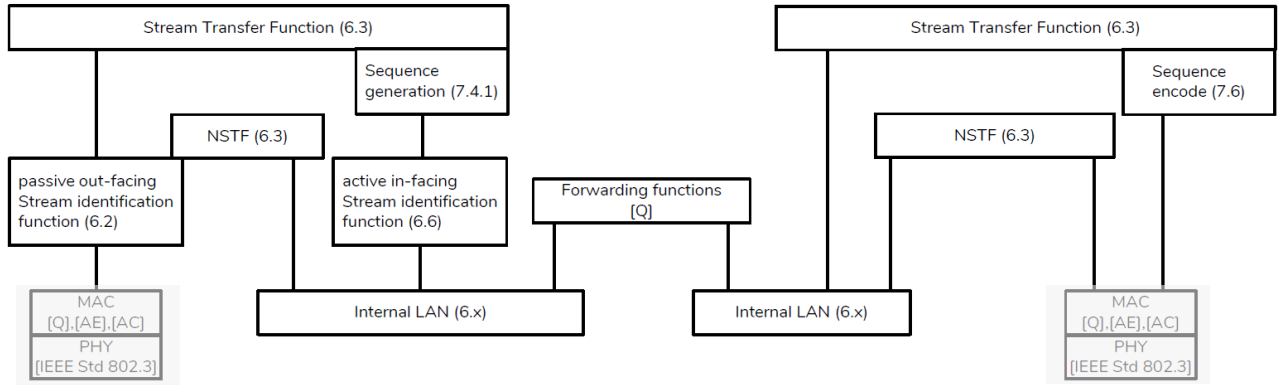
- Introduction of “Lower” and “Upper” Stream identification functions
  - Initial proposed resolution
    - Changes to figures in 6.6



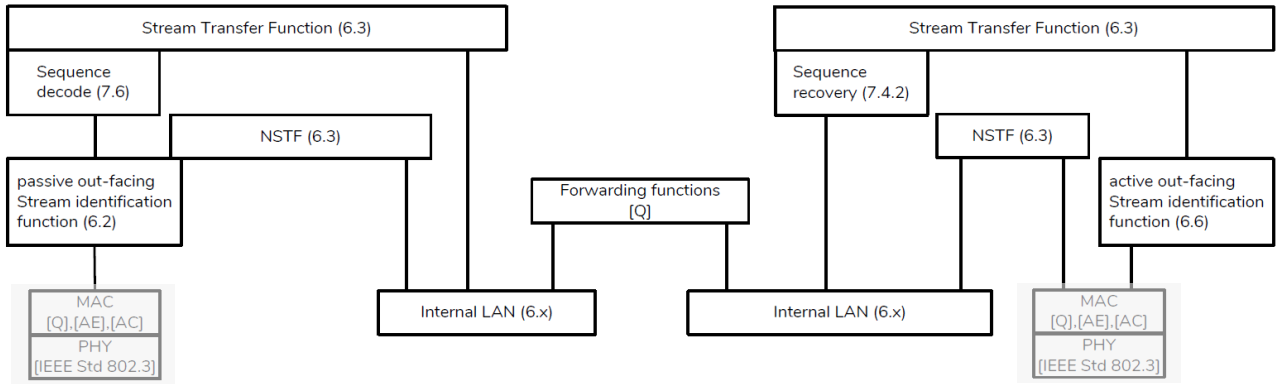
- Introduction of “Lower” and “Upper” Stream identification functions
  - Initial proposed resolution
    - Changes to figures in 8-1



FRER functions in an FRER C-component

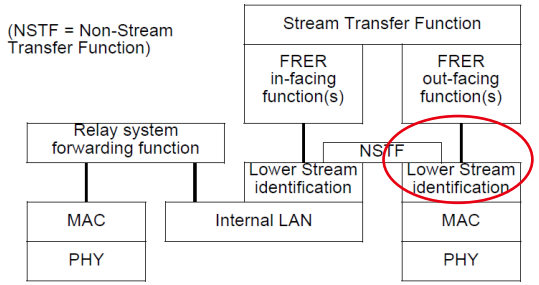


Relay system split

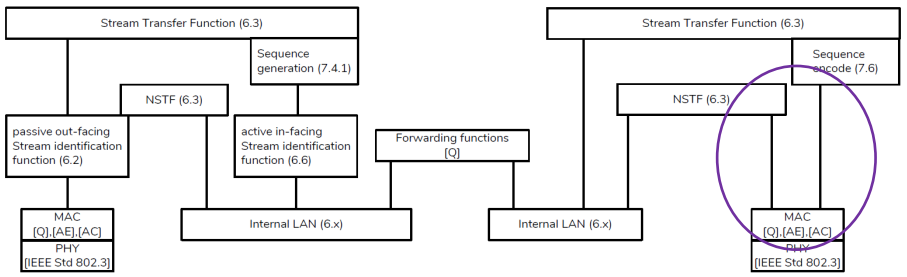


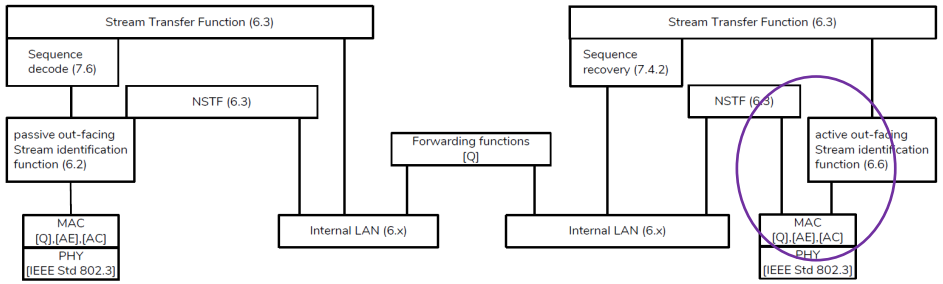
Relay system recover

- Introduction of “Lower” and “Upper” Stream identification functions
  - Editor’s thoughts
    - The lower Stream identification functions « layer » multiplexes the the non-Stream and Stream flows down the stack, towards the MAC.



    - Whereas the schemes below require the multiplexing function to reside in the MAC





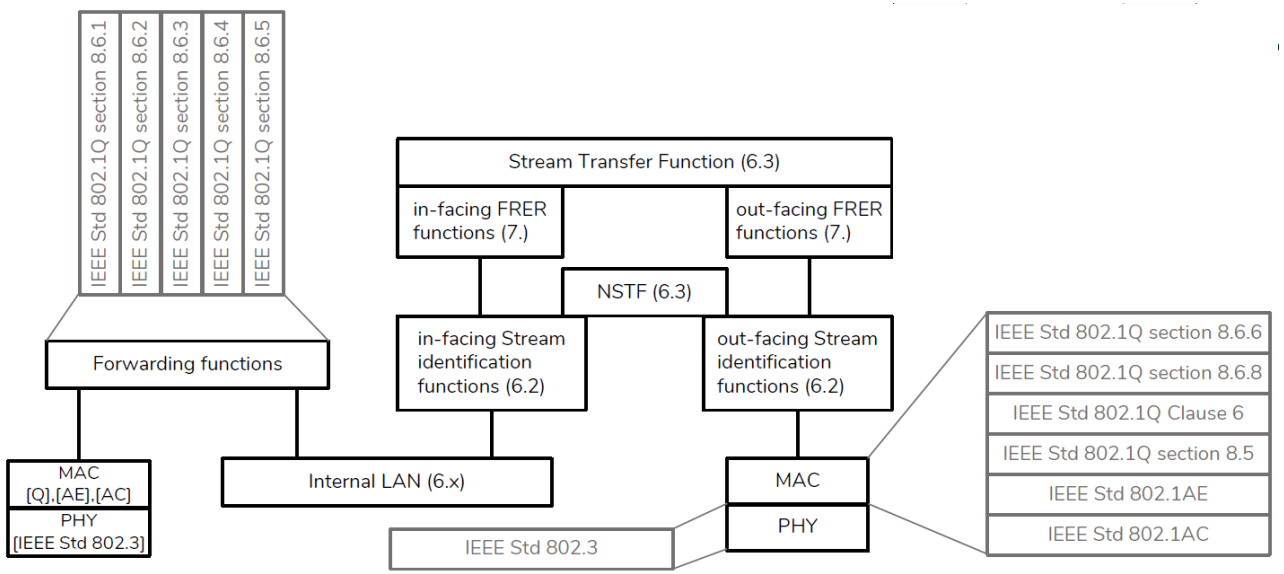
- Difference between the layered model and the “integrated” bridge model
  - Description

    2. Section 8.1 j) “… take place after all forwarding (**except Queuing frames, 8.6.6** of IEEE Std 802.1Q-2014)” and Figure 8-2 contradict Figures 6-5, 6-6, 8-1, C-5, C-6, C-7, and C-12, as the later all show or imply ALL functionality of [Q]:8.6 to be encompassed by the “(Relay system) Forwarding function”.

  - Initial proposed resolution

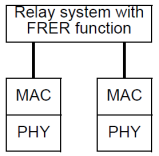
    Since 8.1 j) and Figure 8-2 have been determined to be correct, the other figures need to be adapted accordingly.

- Proposed changes in Fig. C-5,6,7,10,12 consist in introducing Q 8.6.6, 8.6.8, 6, 8.5, AE and AC in the MAC of the port including the Stream identification and FRER function
  - Detailed in the proposed modifications of Fig. 6-6



Proposed Fig. 6-6

- Difference between the layered model and the “integrated” bridge model
  - Editor’s thoughts
    - The layered model is there to permit the introduction of FRER in ports of a relay system, which forwarding function does not support any of the FRER functions, including Stream identification.
    - Figure 8.2 illustrates a way to integrate the FRER functions with the forwarding functions of a bridge
      - Not the same architecture (as the layered model), more a detailed view of Fig. 6-5 a)



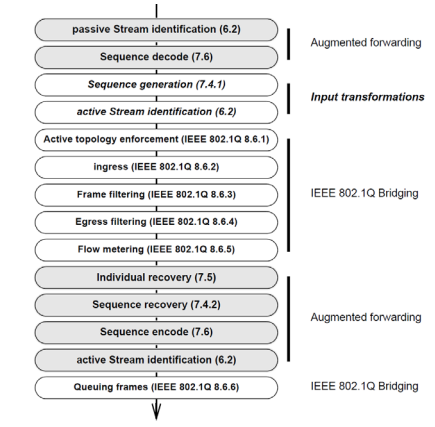
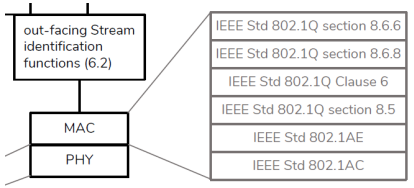
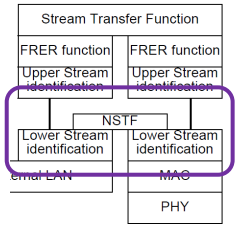


Figure 8-2—Augmented Forwarding Process does sequence recovery

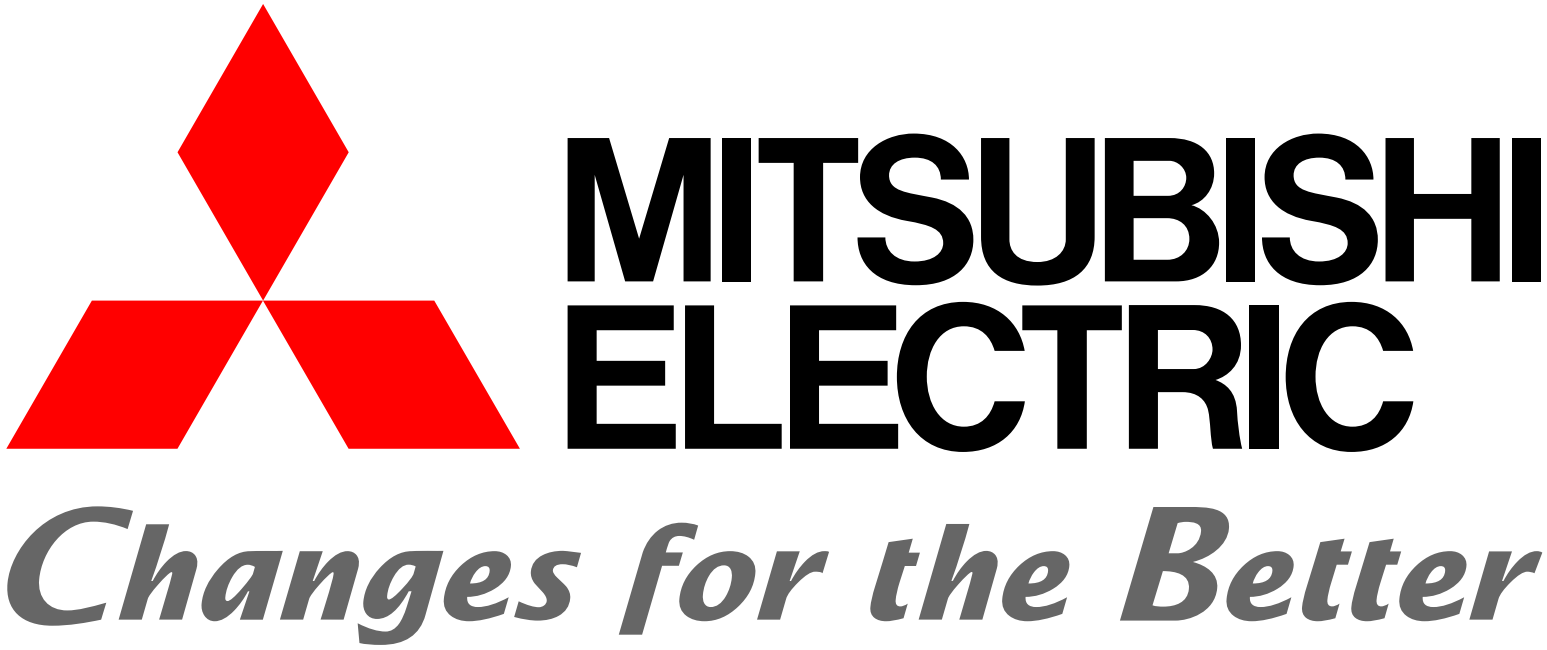
- Part of the forwarding functions should not be « moved » to the MAC (as shown below)



- Question: have the « upper » and « lower » Stream identification functions been introduced as a mean to mux/demux Streams/non-Streams, and especially to multiplex Streams and non-Streams towards the port’s MAC ?
  - Avoiding the addition of queueing functions in the port’s MAC ?



ith separated FRER capability, unting, on an output port.


MITSUBISHI
ELECTRIC
Changes for the Better