

# Seamless Redundancy Issues

## Editor's commentary on P802.1CB Draft 2.2

**Norman Finn**  
Editor, P802.1CB

Ver. 2    March 15, 2016



# Summary of changes in D2.2



# Summary of changes (1/2)

## 1. Shifted to “Member Stream” and “Compound Stream” terminology

But only where these terms are needed because they are relative to each other.

For the most part, just plain “Stream” is used.

## 2. Confluence Sublayer was removed.

But, the “Stream split function” had to remain – that’s the only way the editor could get an end station to transmit packets on both legs of Link Aggregation.

## 3. Source MAC + VLAN Stream identification method was added.

Note that, when this is used, a Source+VLAN “Stream” can easily subsume a number of Streams from a single Talker to different destinations.

# Summary of changes (2/2)

4. The HSR/PRP “match last sequence\_number == Discard” algorithm has been added.

It is defined at a rather low level, parallel to the vector algorithm, because that seemed to the editor to do the least violence to the rest of the document.

5. Autoconfiguration (9.2) was added (see next section of this presentation).

This drove a few changes (bug fixes) in the rest of Clause 9.

6. The (placeholder for the) “Delay function” was removed.

7. Several new clauses were added to informative Annex E:

E.9 Seamless Redundancy and reserved bandwidth (explains Delay function issue)

E.10 Use of the Individual recovery function

E.11 Use of Autoconfiguration

# Autoconfiguration



# Autoconfiguration in D2.2

- Clause 9.2 is “Autoconfiguration”. The ultimate purpose of Autoconfiguration is to automatically recognize packets belonging to new Member and Compound Streams, and instantiate Individual and Sequence recovery functions for them.
- The rest of Clause 9 is largely unchanged, and still supports configuration of CB capabilities on a very detailed basis.
  - Any Stream can be manipulated independently.
  - Every function can be applied to Streams independently, with different parameters.
- Autoconfiguration works by automatically creating detailed configuration entries.
- Why this seemingly complex route? Because the state machines and code fragments are described in terms of the detailed configuration managed objects in clauses 9.3 through 9.7.

# Autoconfiguration explanation in clause E.11

## Four steps to set up autoconfiguration:

1. Set up the routing of Member Streams, especially VLAN translations.  
**This is not an 802.1CB function. It is a relay system function.**
2. Specify on which ports and VLANs packets can be received that will trigger autoconfiguration.
3. Pick the encapsulation to be used on each port when receiving and transmitting autoconfigured Streams.
4. Specify the whether of, and the parameters for, automatic creation of Individual recovery, Sequence recovery, and Latent error detection when packets belonging to new Member and Compound Streams are received.

# Current state

- **Compound Streams** are identified by source MAC address.
- HSR/PRP **Member Streams** are differentiated by the A/B PathId/LanId field in the tag/trailer, not by the port or VLAN on which the packet was received.
- Redundancy tag **Member Streams** are differentiated by the port and VLAN on which the packet was received.
- You can configure Redundancy/HSR/PRP translations without instantiating any state machines.
- Only the match algorithm, not the vector algorithm, can be autoconfigured.
- Because autoconfigured state machines are to some degree ephemeral, only per-port state machine counters are available, not per-port-per-Stream.
- If Port 6 is HSR/PRP Path/Lan A for one autoconfigured output Stream, it is Path A for all autoconfigured output Streams.



# Two Autoconfiguration tables

- The Output Autoconfiguration table specifies, for each port:
  - What encapsulation (Redundancy, HSR, or PRP) to use.
  - If HSR or PRP, whether what constant to use for the PathId/LanID in the encapsulation.
- The Sequence Autoconfiguration table specifies:
  - On which ports/VLANs autoconfigurable packets can be recognized.
  - What sequence\_number encapsulations are allowed, and what to do for each.
  - Whether to instantiate Individual Recovery functions, with what parameters.
  - Whether to instantiate Sequence Recovery functions, with what parameters.
  - When to time out and destroy these state machines.

# The process when a new Stream is recognized

- The process of autoconfiguration, based on a received packet, is in 9.2.2.
- The description is very complex, as mentioned in a NOTE right at the beginning of 9.2.2.
- **Please look at 9.2.2, and see if you agree** that an implementation should be able to create databases and pre-establish resources, before any packets are received, so that the actual creation of per-Stream state machines can proceed quickly and easily, without the elaborate steps described in 9.2.2.

**If you don't agree, please speak up!**. We have a big problem!!!

Why the complex description? Precisely in order to not describe those possible internal implementation databases. The current description depends only on the managed objects in Clause 9.

# Questions

(The following questions were discussed by the TSN TG on March 14, 2016. The editor's opinion of the consensus answers are in **red**.)



# Questions to answer, preferably here and now

## Editor's opinion of consensus answers in red.

1. Do we really need the Stream split function? Is there another way to get Link Aggregation to transmit on both (all) ports? **No answer.**
2. Do we really need to allow multiple input encapsulations for autoconfiguration on the same port/VLAN? On the same port but different VLANs? **Yes (Also, red/blue VLANs Source A = one Compound Stream, green/yellow VLANs Source A = another.)**
3. Should we be able to autoconfigure Compound Streams based on the Destination MAC address, instead of the Source MAC address, thus enabling per-Stream sequence numbering? **Not in next Draft.**
4. Should we allow Member Stream differentiation based on VLAN, instead of on {port, VLAN}? **Not in next Draft.**
5. Should the vector algorithm be autoconfigurable? **Yes.**
6. Should per-Stream counters be available? **Yes, & keep new per-port counters.**
7. Is 9.2.2 (the autoconfiguration flow chart) satisfactory? **No answer.**

Thank you.

