Request for Contribution for Refining the Seamless Redundancy Concept Presented in San Diego

(Version Date: September 4, 2012)

Markus Jochim, General Motors Research & Development
Oliver Kleineberg, Hirschmann Automation & Control

Objectives (1/2)

- During the IEEE 802.1 Plenary in San Diego we presented a proposal for supporting seamless redundancy within AVB2:

  [1] http://www.ieee802.org/1/files/public/docs2012/new-avb-kleineberg-jochim-seamless-redundancy-0712

- Several plenary participants:
  a) made valuable comments after the presentation
  b) expressed interest in supporting this effort!

- Our objective is:
  a) to quickly refine the presented concept for seamless redundancy
  b) to converge towards a refined concept that is widely accepted within 802.1
  c) to get approval for the start of preparation of a Seamless Redundancy PAR at the November Plenary

Objectives (2/2)

- To reach these objectives will require additional interactions / discussion / work in between the 802.1 meetings.
- To start this kind of work, this slide deck summarizes the comments we received after the San Diego presentation.
- Furthermore, this slide deck also provides some initial responses to the comments we received. We hope that this will trigger a fruitful discussion.
- We would therefore like to kindly ask those who are interested in support for Seamless Redundancy in AVB2 to join the discussion / work that we start with this presentation.
- **Important**: We see the redundancy concept we presented in San Diego (as well as this presentation) as a starting point for a discussion that will eventually lead to a proper solution of the problem. The redudancy concept can be refined, tweaked and modified as necessary to achieve this goal.

Scope (1/2)

- The entire problem domain of redundancy (for fault tolerance) for AVB streams can be broken down into two sub-problems:

  P1: Stream path management/configuration

  P2: Redundancy Management
  (= Generation and elimination of redundant messages)

- The San Diego presentation and this slide set are both focused on P2: Redundancy management

- P1 is an equally important topic and will also be handled:
  http://www.ieee802.org/1/files/public/docs2012/new-spbpcr-farkas-SPB-path-control-and-reservation-0712-v01.pdf

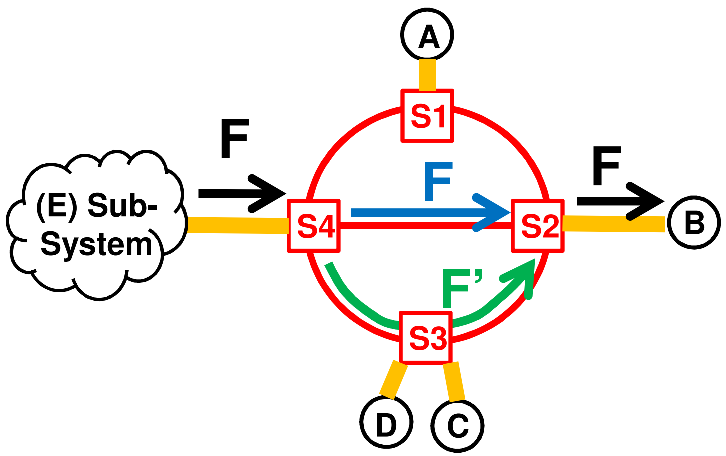
- There are dependencies between P1 and P2.
  Example: Our proposal for solving P2 will result in some requirements for P1 that are discussed on the last slide of this slide deck.

In the following slides the comments we received after the San Diego presentation are discussed.

Comment #1

Comment 1: *"Redundancy management in bridge (Time constraints in fast path)"*

According to slide 25 of [1], the bridge executes the redundancy management function that eliminates duplicates. This is difficult since the function would need to be executed in the fast path and in the fast path there is probably not enough time to execute the function. Evaluate the option to execute the redundancy management function in the listener instead!



Response to comment #1

– If the listener executes the redundancy management function, both messages F and F’ have to be forwarded from S2 to B. The link may become be a bandwidth bottleneck.

– Depending on the complexity of the network and the number of redundant paths, the bottleneck problem may get worse and network design flexibility can suffer.

– Even if integrating redundancy management functions into the bridges is challenging, lightweight duplication elimination techniques (like a multiple sliding /drop window algorithm) could be used – especially if seamless operation is only considered for scheduled traffic

We believe it is worth to start out by collecting some opinions on which solutions within bridges might work.

Comment #2

Comment 2: *“Switching over from stream F to F’ ”*

a) For rate constraint traffic, slide 26 of [1] anticipates that F and F’ should be considered to be a primary and a secondary stream. How can the bridge S2 tell the difference between a) a scenario where F is unavailable (due to a fault) and b) a scenario where temporarily no stream F data is received since the talker has currently nothing to say?

b) Also: There is a risk that switching over from F to F’ could result in a scenario where during the transition phase some frames might be received by B either twice (once from F and once from F’) or not at all for a very limited time.

Example used to illustrate the issue: (Please refer to the diagram on slide 26 of [1])
F = primary stream, F’ = secondary stream. Assume that the frames that belong to stream F (blue) arrive at S2 earlier than the frames that belong to stream F’ (green). If F becomes unavailable, a fast switch-over to stream F’ can result in a situation where S2 receives some frames on stream F’ that S2 already received on stream F before F became unavailable. Result: Reception of duplicated messages for a very limited time during the transition phase. Similar example that results in loss of frames for a very limited time can be constructed if frames from F arrive at S2 later than frames from F’.

Response to comment #2

– We discussed different ways of how the issues pointed out in comments 2a and 2b could be addressed.

  For example:
  If S2 "knows" the maximum delta_t, by which the arrival times of frames from both streams F and F' may differ in a fault free case, then S2 can use max_delta_t as a timeout.

  If after max_delta_t one a frame from F' is not matched by a redundant frame from F, than this indicates that F has become unavailable. After a configurable number of consecutive missing frames from F, the stream F is assumed to be unavailable or at least unreliable. This triggers the switch-over to F'.

– However, we came to the conclusion that we should first discuss if a seamless redundancy concept really needs to be available for rate-constraint traffic. If we come to the conclusion that seamless redundancy can be limited to scheduled traffic only, we can avoid all the complexity involved with solving the issues pointed out in 2a and 2b.

Comment #3

Comment 3: “Different link speeds”

a) In the diagram on slide 24 of [1] the frames F and F’ may arrive at S2 at different points in time. How do we address this problem especially in scenarios where the frames arrival times may differ quite a bit (e.g. because the links in the topology are operated at different speeds).

Response to comment #3 (Slide 1/2)

- One possible constraint introduced could be the restriction of only allowing redundant paths to have the same link speeds.

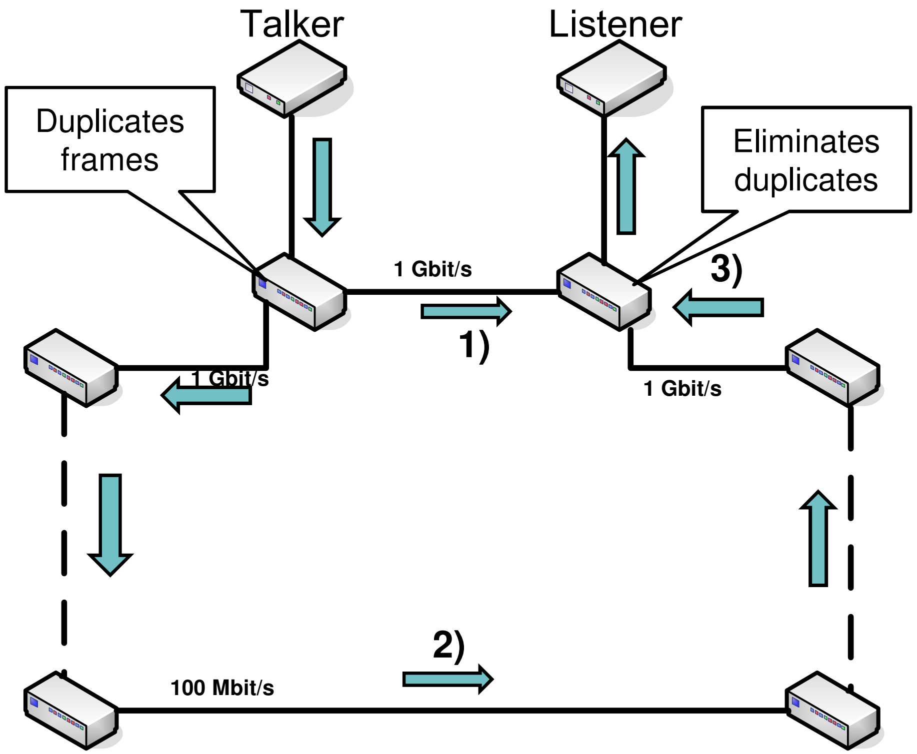
- There will always be a certain difference between the frames from F and F‘… the redundancy management (in this case the duplicate elimination procedure) must be designed in a way that it can detect duplicates within the worst case time difference between F and F‘

If we assume that differing link speeds on different paths, in respect to redundancy, result in different max. latency, the problem is reduced to handling the time difference between the arrival of F and F‘. Bandwidth implications will be handled by SRP.

Response to comment #3 (Slide 2/2)

Depending on placement of sender and receiver in a redundant link network and depending on the link speed of individual segments, there can be considerable time skew. Redundancy management needs to take this time skew in consideration, as described in this example of a single redundant ring network with asymmetrical placement of talker and listener:

1) **At t_1, the first frame of a duplicate pair (in this case it is a pair, because two paths are configured) arrives at the receiving bridge**
2) **The second frame of the pair has to travel a significantly longer way route and experiences higher latency**
3) **At arrival time t_2 of the second frame, the bridge still needs to be able to identify the duplicate after delta_t = t_2 – t_1**



Comment #4

Comment 4: *"Identifying frame pairs based on arrival times"*

For scheduled traffic, slide 26 of [1] states that the identification of frames that form a redundant pair should be based on the expected arrival time. An alternative solution would be to use sequence numbers within the frames.

Response to comment #4

"Identifying frame pairs based on arrival times" – Pro's and Con's of pure Arrival time vs. pure Sequence number

Arrival time	Sequence number
(-) Works only with scheduled traffic	(+) Works with scheduled and rate-constrained traffic
(+) Tight duplicate elimination scheme within a single time slot - results in simple implementation schemes	(-) Duplicate elimination not restricted to time windows – more complex implementations needed
(+) Aligns well with concept that all redundant frames regardless of communication path need to arrive within a certain time window	(+) Aligns well with concept that all redundant frames regardless of communication path need to arrive within a certain time window
(+) Reduction of scope to scheduled traffic makes the algorith hard to beat	(-) Algorith can be beaten if delta_t between arrival of the two frames forming a pair is too high (impl.specific)
(+) No impact on frame format.	(-) Frame format needs to accommodate for a sufficiently large sequence counter.

An alternative is to combine the two concepts for scheduled traffic – this results in a small sequence number counter for the scheduled traffic phase

Request for contributions

– We are looking for comments and contributions to refine the seamless redundancy concept. Please respond via reflector / email.

– Depending on the responses we may also set up a conference call to further discuss the topic. A discussion during the AVB conference call is also a possibility.

– If you are interested in contributing / participating in the conference call, please indicate your interest, so that we can invite everybody interested!

Appendix: Stream path management related requirements

The solutions discussed in this slide deck include some assumptions that result in requirements towards stream path management:

- It is implied that stream path management also factors in that (in case of scheduled traffic), latency on all configured redundant paths does not exceed a known limit that is defined e.g. by an application as a maximum latency that can be tolerated
- This, in the following, implies that the approach does factor in network topology as well as link speed and aggregates this information to a single max. path latency value that can be evaluated against the end-to-end max. latency requirement between talker and listener