

Quick Failover Algorithm in SCTP

draft-ietf-tsvwg-sctp-failover

Y. Nishida, P. Natarajan, A. Caro,
P. Amer, K. Nielsen

IETF92, Dallas

Status

- -08 for WGLC 29.10-19.11.2015
 - Technical comments received from Martin Becke and Rob Braden.
 - Post WGLC review (on -09 24.12.2014) performed By Gorry Fairhurst
 - 09 not adequate to address all WGLC comments [PFMR=0 does not disable]
 - Further issues on standard language + request for additional elaboration on the motivation for certain standard language requirements
 - RFC4960 Update flag Issue
- -10 submitted 09.03.2015 to address the above
- Differences -08 to -10:
 - Clarifications. No “intended” change of function, “except” still changes of standard language.
- Michael Tuexen review on version 10:
 - Issue for security section, remove cc text description, HB transmittal in PF state, dormant state
- Self-comment on Standards Language om api events in -10

Version -08 to -10 Clarifications

- **WGLC: Why does PFMR=0 not mean SCTP-PF logic off. New text clarifies:**
 - PFMR does not control activation of SCTP-PF logic
 - PFMR controls PF state threshold for reach of PF state
 - PFMR \geq PMR means no PF state for destination address
- **HB transmittal during PF state:**
 - HB MUST be sent per RTO in PF state
 - Text un-clarity on whether RTO back-off is done or not (yes) addressed by using RFC4960 formulations (no keywords)
- Michael Tuexen comment on -10: For consistency HBs in PF state should honor user configuration. **Proposed resolution:**
 - HBs in PF state MUST honor user configuration for RFC4960 HB
 - HBs SHOULD be sent per RTO in PF state

Michael Tuexen Comments

- Document Security Issue of PF
 - PFMR=0 makes it easier for an attacker to make SCTP MH change path
 - Even more forceful with Permanent Failover
 - To adopt in next version
- Enforce Dormant state as non-SCTP PF specific feature:
 - SCTP continues to transmit in dormant state
 - SHOULD for SCTP-PF because SCTP-PF should not compromise disruption tolerance compared to RFC4960
 - Feature also beneficial for RFC4960 SCTP tolerance -> propose as feature in its own right.
 - To adopt in next version
- Remove congestion control language (even if no change intended of RFC4960) - to adopt in next version

Standards language on API

- -10 says:

10. The SCTP stack **SHOULD** provide the ULP with the means to expose the PF state of its destinations as well as the means to notify the state transitions from Active to PF, and vice-versa. When doing this, such an SCTP stack **MUST** provide the ULP with the means to suppress exposure of PF state and associated state transitions as well.

- To be changed to (no standards language on API)

10. The SCTP stack **should** provide the ULP with the means to expose the PF state of its destinations as well as the means to notify the state transitions from Active to PF, and vice-versa. When doing this, such an SCTP stack **is recommended to** provide the ULP with the means to suppress exposure of PF state and associated state transitions as well.

RFC4960 Update flag ?

- This PS complements, but does not update, RFC4960
- No comments have been received to the contrary – but question raised by wg chairs.
- Can we consider this issue as being resolved ?

Proposed Next Steps

- All technical issues raised , including standards language, are (believed to have been) resolved.
- BUT... some re-structuring would still be beneficial given PS
- Next step would be a -11 revision:
 - Implement the agreed technical resolutions
 - Still improve language and some restructuring
- Wg chairs to decide if second WGLC is required
- Hopefully then ready for the IESG (before IETF93?)