

# Quick Failover Algorithm in SCTP draft-ietf-tsvwg-sctp-failover

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

IETF 89, London

# Status

- Version-03 uploaded 02.03.2014
- Draft has been updated in accordance with decisions taken at IETF 88 in regard to all known open points:
  - CNWD handling in PF
  - Association Error Counter Handling
  - Permanent Failover
  - APIs for PF

# CWND handling in PF

- PF is kept independent from congestion control
  - Suggestions related to cwnd/ssthresh have been removed

# Association Error Counter Handling

- PF state exposes generic problem with counting of HB failures in association error counter when the HB rate is different on the different paths
- Generic RFC4960 issue handled by Errata 3788 (Verified).

# Permanent Failover

- Purpose of Quick Failover is to improve the failover performance of SCTP. Also the switchback operation after failover significantly impacts the performance.
- Permanent Failover switchback operation now adopted (MAY) in draft.
- [RFC4960] switchback behavior is suboptimal in certain situations, especially in scenarios where a number of equally good paths are available. It is recommended for SCTP to support also, as alternative behavior, the Permanent Failover modes of operation where forced switch back to a previously failed primary path is not always performed.
- We recommend that SCTP-PF should stick to the standard RFC4960 behavior as default, i.e., switch back to the old primary destination once the destination becomes active again. However, implementations **MAY** implement Permanent Failover and **MAY** enable it based on network configurations or users' requests.

# APIs for PF

- API for control of PF feature
- *NEW*: API for control of switchover mechanism
  - RFC4960 Default or Permanent Failover
- *NEW*: API for notification of PF state changes:
  - ACTIVE → PF, PF → ACTIVE
  - via existing SCTP\_PEER\_ADDR\_CHANGE, SCTP\_GET\_PEER\_ADDR\_INFO
- *NEW*: API for control of whether PF state changes are suppressed (to support legacy RFC4960 state machine, to support ULP which don't care).

# Next steps

- Experimental or PS ?
  - Authors propose for PS.
  - Not sure what we will do for experiments
  - Logic is pretty simple
  - Analysis has been done in several papers
  - Function running in deployment
  - Already have three implementations (FreeBSD, Ericsson, Linux)
- Progress to WG LC ?