Agreement Protocol for Synchronization

Application Level

- M, ΔM

  Membership related, see "Usage of A.P. ..."

- T_{sync}

  Sync topology (who syncs with who)

  Examples: Per destination, all flows
  Voting may be desired (e.g., would you accept sync with Portugal?)

- P_{sync}

  Sync policy (e.g., how quickly to adapt)

  State S_i: Sync function (sync agent)
  $S_i \in E$: (Eventually) consistent
  Up: Convener
  Implement: Poll (lock) announce commit, (release)

Synchronization Initialization

Initialization

Message from "convener" sets session-wide parameters (e.g., epochs for updating common sync delay)

- S_{init}
  All initialization message parameters
- Policy
  Set/change initialization parameters

  Voting: None

  Consistency:
  $S_{init} \subset I$ (states affected) $= E$ (only global parameters)

  $\mho = \mho_{init}$: Only convener/initiator can set S_{init}
- Implement
  "Announce" or: (lock), commit, (release)

Agreement Protocol for Synchronization

Adapt to Flow Delays

Adapting

Periodic message communicate (changes in) flow delays

- S_{dest}
  Flow parameters (e.g., delay, delay tolerance, priority)
- Policy
  Regular updating of flow parameters

  Voting: None

  Consistency: $S - E$ (S_{dest} not globally observable)

  $\mho = \mho_{dest}$
- Implement
  "Announce"

  [L.S.A.: (Poll;) Announce]

Food for Thought

- Should A.P. allow "scheduled votes" (e.g., in periodic adaptive policy)?
- Should A.P. allow "non-global states" in set E (e.g., Local agreement within subsets of M)?
- Should "Adapting Sync Delay" be considered an agreement?

  If not, how does one recognize "agreement"?

  If so, should votes/results allow numerical value (e.g., how long should we spend on this)?