

P802.1DD d2.0

Proposed Resolution for Comment 19

Martin Mittelberger

Siemens AG

2026-09-15

v03

Comment 19 (P34L1)

Comment:

Because RAP commits resources hop-by-hop before the end-to-end reservation outcome is known, a reservation that ultimately does not succeed can temporarily consume resources needed by another reservation that could succeed. This is an early resource commitment versus tentative admission followed by resource commitment. If there are cases where the unsuccessful reservation is repeatedly attempted, the first-come-first-served rule for equal-Rank streams appears capable of repeatedly granting those resources to the unsuccessful reservation, potentially starving an otherwise feasible reservation.

Suggested Remedy:

The RAP allocation description should describe the reason for choosing early resource commitment versus tentative admission followed by resource commitment. Then the draft should describe the mechanism, if any, that guarantees convergence or fairness in first come first served cases and should discuss whether repeated unsuccessful reservation attempts require retry/backoff or other protection.

Background

- Because RAP commits resources hop-by-hop before the end-to-end reservation outcome is known, a reservation that ultimately does not succeed can temporarily consume resources needed by another reservation that could succeed.
- This early resource commitment is intentional and follows the same principle as MSRP ([Q] 35.2.2), avoiding the additional protocol round-trips, state, and latency that a tentative-admission-then-commit scheme would require.

Regarding fairness among equal-Rank streams:

- the relative importance ordering defined in 6.3.11 (reservationAge, then StreamId, then VID) is fully deterministic.
- Since no two reservations on a Port share the same <StreamId, VID> pair, this ordering always yields a strict, unambiguous result — there is no scenario in which the ordering is undefined or arbitrary among equal-Rank reservations.
- Handling of repeated unsuccessful reservation attempts, including any retry or backoff strategy, is a matter for the RAP Endpoint user or network management (see 6.5.2) and is outside the scope of this standard, consistent with existing MSRP practice.
- RAP intentionally reuses the MSRP-established approach in this area.

Comment 19 (P34L1)

Proposed Response:

PROPOSED ACCEPT IN PRINCIPLE.

Add a note in 6.1 (RAP overview):

“RAP commits resources on a per-hop basis as the Talker Announcement and Listener Attachment propagate through the network, rather than using a separate tentative-admission phase prior to commitment. This follows the same principle as MSRP ([Q] 35.2.2) and avoids the additional protocol round-trips, state, and latency that a two-phase admission scheme would require.”

Add a note in 6.3.11 (Stream rank and reservation importance):

“The relative importance of reservations of equal Rank is fully deterministic. The rules in this subclause (reservationAge, then StreamId, then VID) always yield a strict, unambiguous ordering. Reservations are handled on a first-come-first-served basis consistent with existing MSRP practice. Handling of repeated unsuccessful reservations, including any retry or backoff strategy, is a matter for the RAP Endpoint user or network management (see 6.5.2) and is outside the scope of this standard.”

Add the following note in 6.3.7 (Talker announcement):

“A Talker Announce registration results from a single ANNOUNCE_STREAM.request (6.5.2.2.1) issued by the RAP Endpoint user. RAP itself does not automatically retry or re-issue a Talker Announcement following an unsuccessful reservation attempt. A new Talker Announcement for the same StreamId is only created as a result of an explicit new ANNOUNCE_STREAM.request by the RAP Endpoint user, for example following a reset or a configuration change of the Talker.”

Questions?