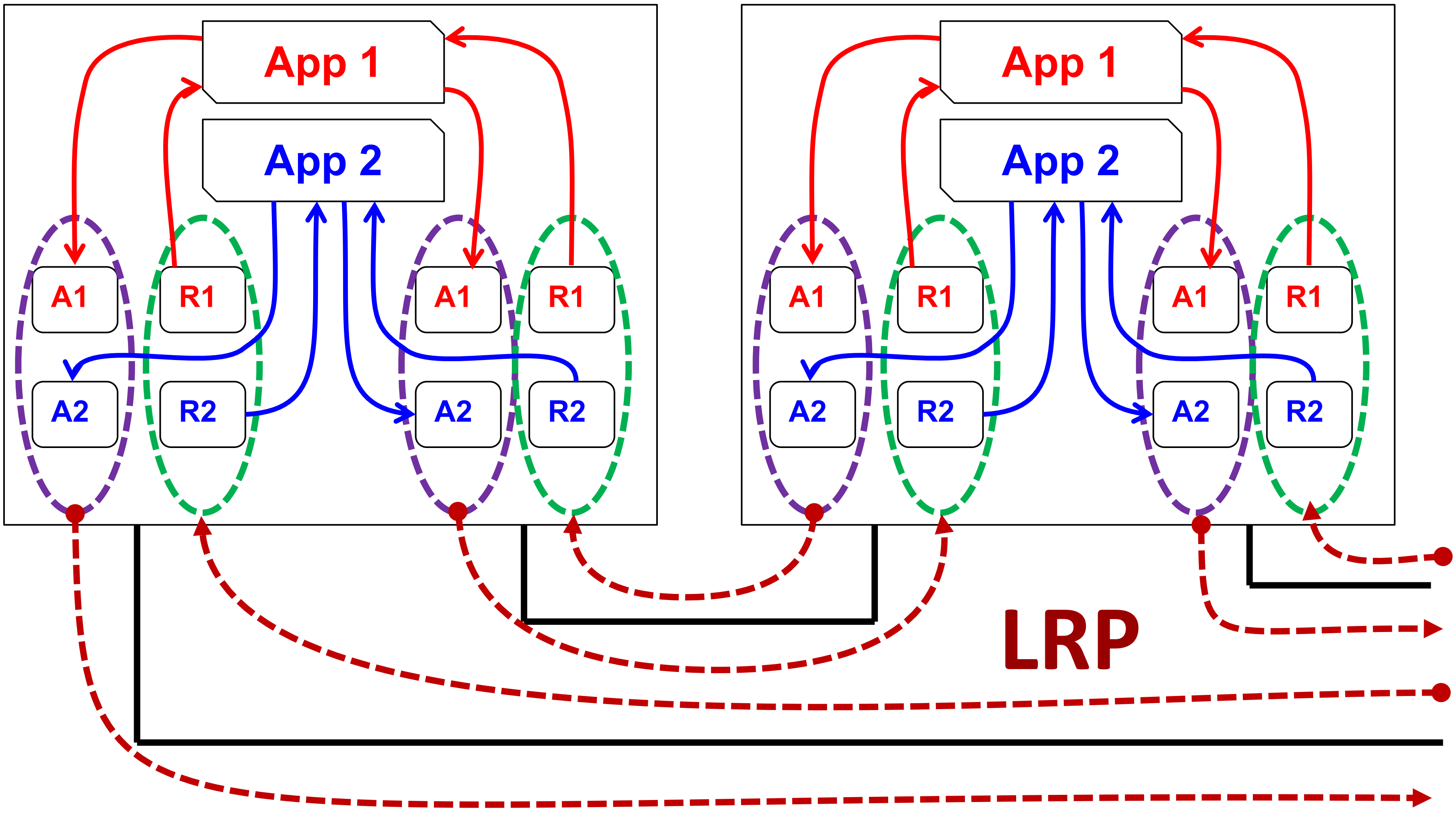
LRP Draft 0

Norman Finn, May 2017

Draft 0

- There are many approaches to this problem. Two were outlined in cs-finn-link-registry-options-03-17-v02. In the absence of further guidance from the committee, I chose the “IS-IS-like” option for LRP_{R0}.
  - This works more like MRP than the TCP option.
  - This is more limited in its potential scope (no direct Talker-Controller option to create).
  - It is much more efficient than MRP (to be explained).

Applications, databases, and LRP



LRP_{R0} vs MRP: Similarities

- Each application has 0 or 1 applicant databases and 0 or 1 registrar databases per port. Multiple applications are supported.
- The application controls its applicant databases on every port.
- The registrar databases are inputs to their applications.
- LRP_{R0}/MRP promptly and reliably replicates the applicant database to the registrar database at the other end of the link.
- There are primitives linking LRP_{R}/MRP to the application.
- Both do keep-alive.

LRP_{R0} vs MRP: Differences

- LRP_{R0} is optimized for carrying ~ 1 Mbyte across a link, where the database consists of some number of Records (Blocks).
- MRP is designed for carrying ~ 1.5 Kbytes across a link, where the database consists of a list of approximately-consecutive integers.
- MRP is optimized, on a shared medium, for a registrar database that is the union of all other application databases on the medium.

LRP_{R0} vs MRP: Differences

- LRP_{R0} can carry any number of applications in one LRPDU. MRP carries only one application in an MRPDU.
  - MRP uses only application per PDU because it uses a different destination address for each application. This allows unsupported applications to pass transparently through non-participating bridges.
  - But, this makes every link potentially a shared medium.
  - LRP_{R0} can carry any number of applications because it is limited to point-to-point links, and therefore every app has the same reach. (See multipoint discussion, below.)

LRP$_{R0}$ vs MRP: Differences

- LRP$_{R0}$ is based on IS-IS (ISO 10589). Four PDU types:
  - Hello: for neighbor and application discovery
  - Record Transmission (IS-IS LRP): for sending application data
  - Complete Sequence Number: for consistency checking
  - Partial Sequence Number: for acknowledging data transmissions
- Acknowledgement, periodic revalidation, and keep-alive are all handled by complete database retransmission in MRP, and by checksum exchanges in LRP.
- LRP$_{R0}$ supports different destination MAC addresses for different (point-to-point) reaches, e.g., via provider bridges.

Shared media

- If we decide to support shared media == support non-participating nodes, more work is required.
- An IS-IS-like protocol (or a TCP-based protocol) would presumably establish *n*(*n*-1)/2 point-to-point relationships on among *n* peers on a shared medium.
- If we can resolve this quickly, it will greatly facilitate the progress of P802.1CS.
- This would make it more complex to share an LRPDU among multiple applications.

Shared media

There is a fundamental difference between IS-IS and MRP behavior on a shared medium.

- LRP$_{\text{IS-IS}}$ would maintain a registrar database that keep a separate list of Records for each applicant on the medium.
- MRP maintains a registrar database that is a union of all of the applicant databases on the medium, irrespective of which applicant(s) sourced a given integer.

Security

- MRP depends on MACsec.
- LRPR0 makes no mention of security, so by default, depends on MACsec.
- IS-IS security is defined by IETF, and might be applicable to LRP. I have not investigated how this works.

Questions

This project can progress only as these questions are answered. This editor will not answer them.

1. Should the registrar database be a union of applicant databases or keep separate data? (This editor is not sure how to do the union with IS-IS.)
2. Do we do an IS-IS-like LRP or a TCP-based LRP? (If anyone wants to do an MRP-based LRP, then devise and present that solution to the group. Note the issues raised in this preso.)
3. Do we support shared media == pass-through nodes?
4. Is MACsec sufficient, or should we adopt IS-IS security?

Thank you