

RIFT Auto IS-IS

(was Auto-Flood Reflection)

draft-head-rift-auto-is-is-00

IETF 121

Dublin

Jordan Head

(presenting)

Tony Przygienda

Colby Barth

Luis Tomotaki

(new)

First, a quick refresher

- IS-IS Flood Reflection helps improve scale vectors in large single area Level 2 networks.
 - e.g., Over flooding, LSDB size, SPF computation, etc.
- Flood Reflection topologies were in a sense, a convenient first use-case for us to implement.

Why the name change to Auto IS-IS?

- Many networks provision more generic IS-IS topologies (e.g., L1, L2, or a mix of both).
- The document was ultimately restructured to treat “Auto-Flood Reflection” functionality as a “flavor” of what is now Auto-IS-IS.

And now a disclaimer

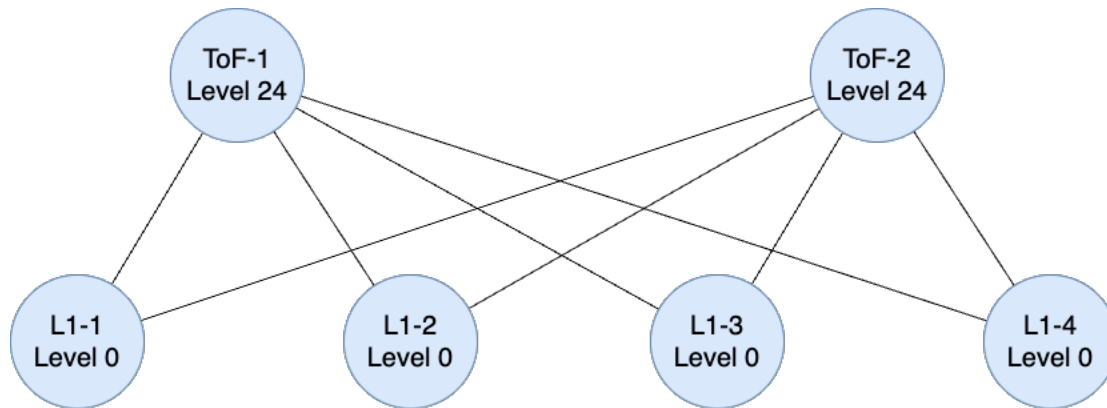
- Everything in this specification leverages *existing* IS-IS standards.
- In other words, we are not changing *any* IS-IS behavior with this specification.
- We'd be happy to share with LSR if the WG thinks it adds value.

What is Auto IS-IS?

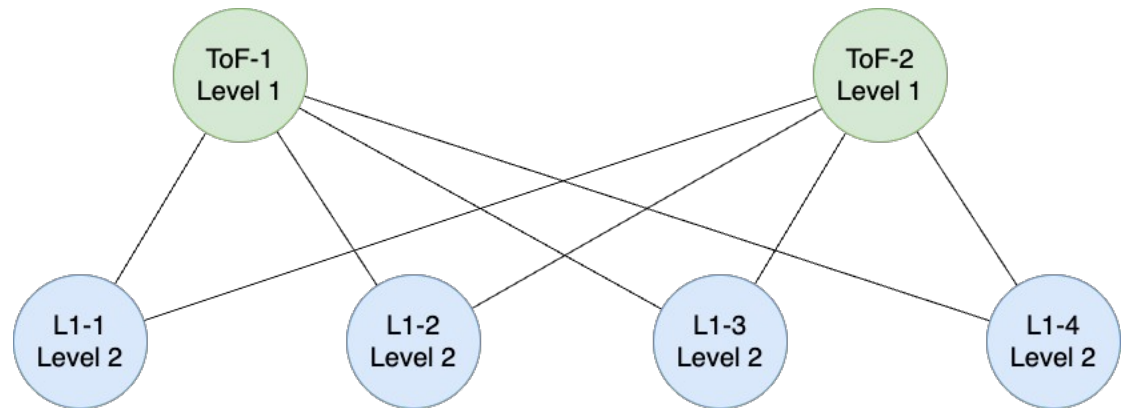
- ZTP brings up the RIFT fabric while LIEs signal a node's capability and desire to self-configure in an IS-IS topology.
- Normative algorithms perform distributed computation to derive necessary variables:
 - Loopback address
 - ISO System ID
 - NET Address
 - etc.
- Roles (within the LIEs) will enable the ability to determine what level a node or its interfaces should participate in.
- RIFT Key-Value TIEs distribute fabric information for simpler operation.

Let's visualize the whole solution.

1. The ZTP process derives each node's level in RIFT.

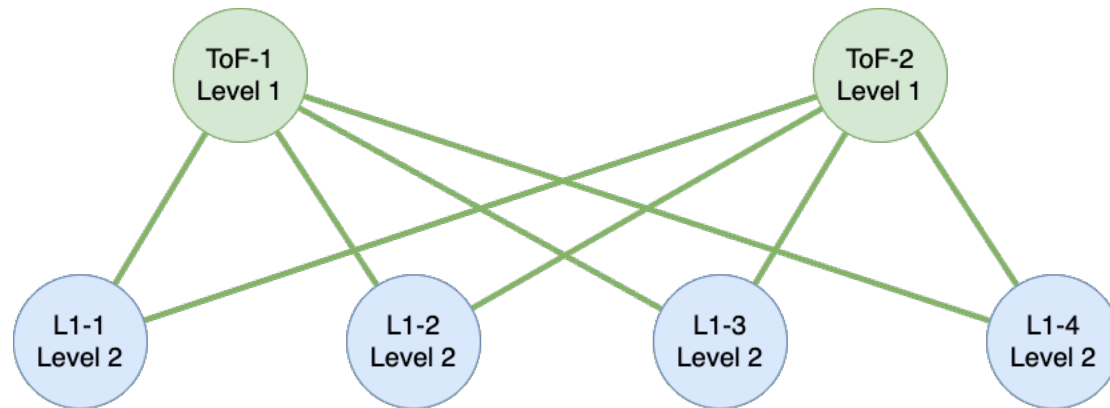


Let's visualize the whole solution.



1. The ZTP process derives each node's level in RIFT.
2. LIE FSM helps derive necessary IS-IS variables for self-configuration.

Let's visualize the whole solution.



1. The ZTP process derives each node's level in RIFT.
2. LIE FSM helps derive necessary IS-IS variables for self-configuration.
3. IS-IS topology comes up.

What's the goal?

- One of the primary drivers of RIFT was that you could just “plug it in” and everything would just work.
- With networks commonly leveraging IS-IS to do chassis disaggregation, why not apply the same logic?
- We've already achieved this with Auto-Flood Reflection, but generic IS-IS topologies are more widely applicable.

What else is new?

- **Normative Changes**

- In the context of Auto-Flood Reflection, the previous version indicated that spine nodes in a Flood Reflection topology did not have to support Auto-Flood Reflection. We changed that to “MUST” as spines need to understand their place in the fabric so they can self-configure as L1 nodes.

- **Editorial Changes**

- References, etc.

What now?

- With the new charter ratified, we are asking for Working Group adoption.

What's next?

- Normative algorithms
 - We effectively have these already as Auto IS-IS will use most of the same variables as Auto-Flood Reflection
- Key Value analytics
 - Similar considerations as above
 - Will likely add more to cover other operational considerations.
- Role definitions for level determination
- YANG model
- Share with LSR?

Questions?