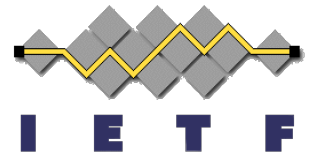
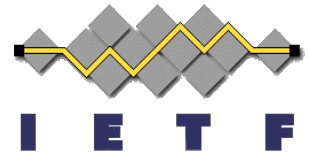


AD-RIFT
draft-przygienda-rift-adrift-00
IETF 120

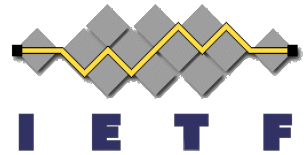
T. Przygienda
Juniper Networks





Let's chase the AI fad

- AI surely an aspect but it's basically TE meets RIFT
 - Called “Adaptive” routing now
- Problem has 2 major aspects
 - We need to flood bunch TE metrics around
 - We have to figure out how to steer northbound without ending up dead at scale back in flat flooded IGP land

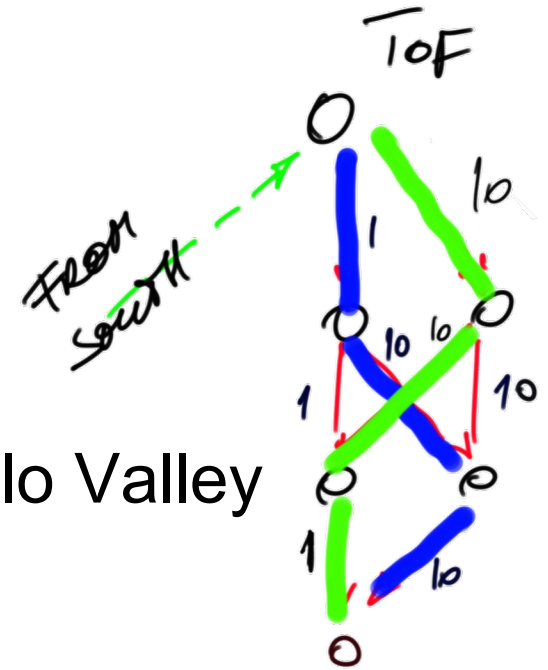


The annoyingly trivial part

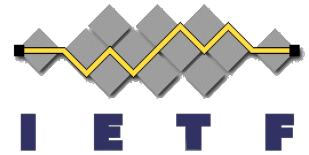
- Add TE metrics to RIFT
- Sticking a new TIE type in: `_AdditionalNodeTIEElement_`
- Per link some general TE metrics and per traffic class metrics
 - BW is already in RIFT but it's cumulative in parallel link cases
- Low priority flooding
- Node TIE scope
- Southbound full view of metrics
- Northbound the usual, 2 hops horizon
 - Anything else becomes flat IGP

Computation

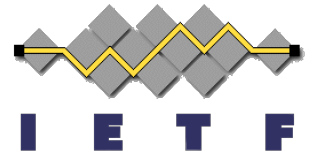
- Southbound is trivial since full topology visible so we just adjust weights on next-hop gateways in RIFT
 - That's a brazen lie of course
 - Classical fishtail problem here
- Thankfully we have TEAS
 - CSPF Problem = RSVP-TE Modulo Valley Free Routing



Computation for the interesting part



- Northbound we need “coarse congestion control signalling”
- TEPrefixPreferenceTIEElement is a new TIEType
 - Prefix Scope
 - Equivalent to negative but with `relative preference to keep traffic off a node`
 - Generated based on checking how good is the southbound path from a ToF compared to others at same level
 - Can be transitive and generated at any level
 - Computation follows negative disaggregation computation and adjusts remaining gateways on the nexthop
 - Negative and preference computation can both generate more specifics of course but preference must respect negative disaggregation results, i.e. if a prefix is fully negatively suppressed then the even more specific preferred cannot forward



Other Stuff

- Very annoying, new major schema version will be needed
 - New TIEType which has different scope than Prefix TIE
- Routing will never do more than “coarse” balancing
 - Otherwise it’s flow-based forwarding, never scaled, never will
 - Destination source-specific congestion signalling will be always much faster than all the routing machinery
 - Mix of both needed to scale, shift load on coarse scale by routing, patch up remaining stuff/transients by CAC
- Draft needs more details on computation and so on, co-authors welcome