

Routing in Fat Trees (RIFT) Update draft-rift-rift-04

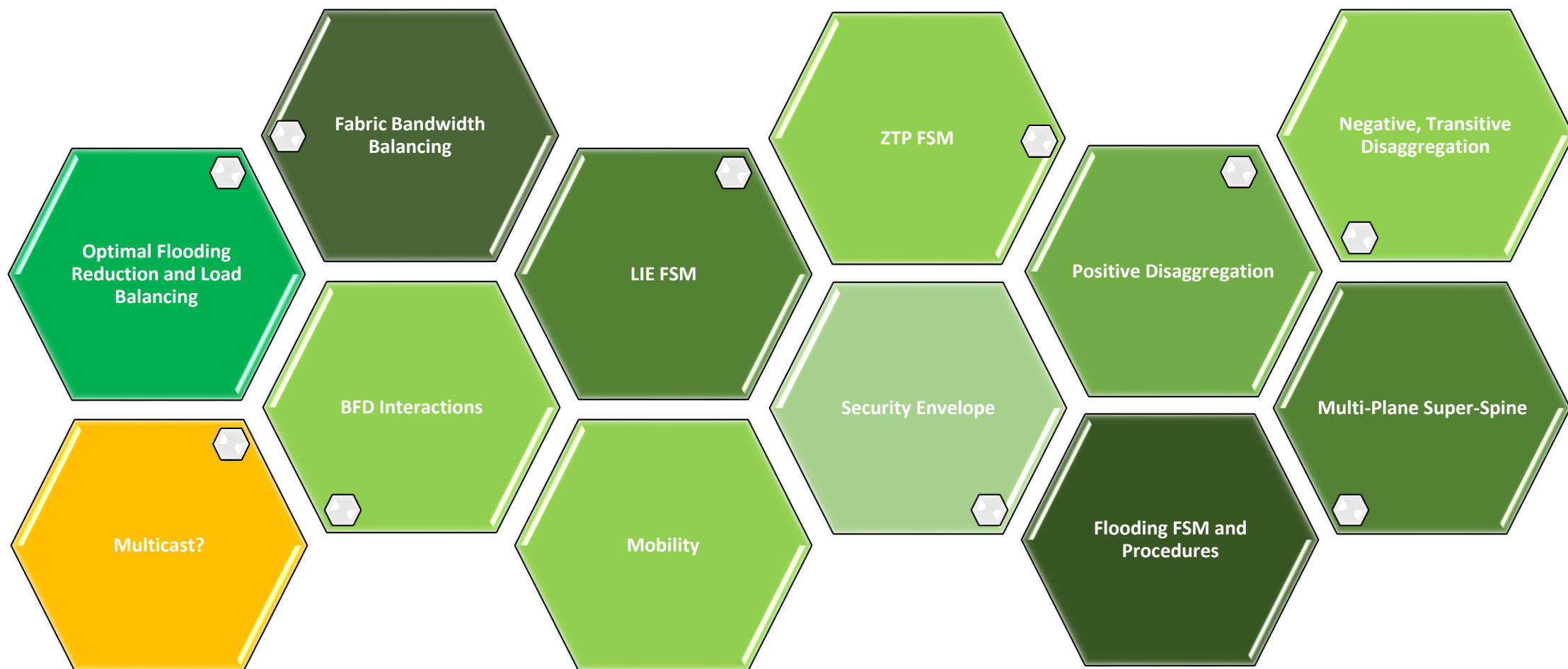
IETF 104, 3/19, Prague

The RIFT Authors

Update from -03

- Last version presented in Montreal -03
- We went to -04 since then
- Specification Holes pretty much all closed now
- Lost of Open Source Code written and interop'ed
- Once or twice weekly online meetings has been held by the 'core crew' on ongoing basis
 - Most meetings recorded and posted to mailing list

Update -03/-04, Green is Done



Rough Statistics

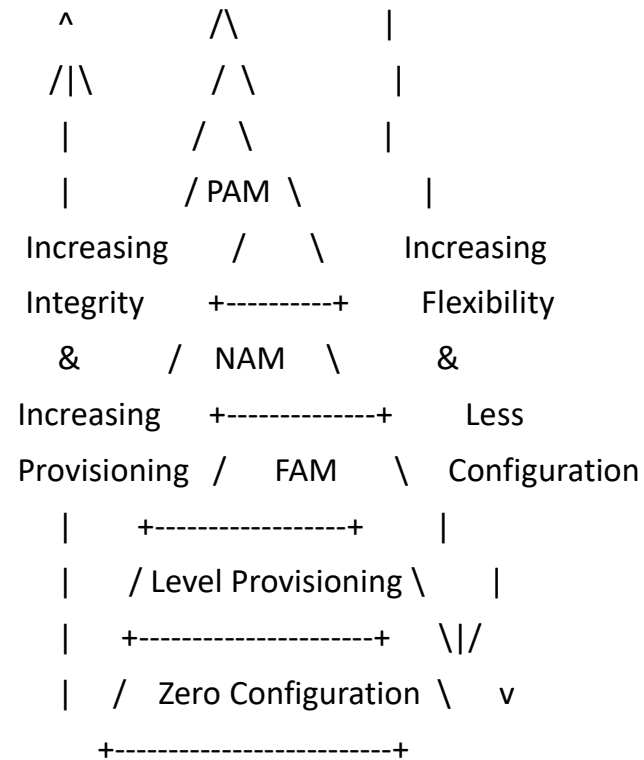
- Emails on “core contributor” email threads since last IETF: hundreds
- Commits on Open Source version since last IETF without branch merges: 380 (last time was 205)
- Lines on Open Source version patch since last IETF: 24’622 (last time was 15’897)
- Diff Size Between -03 and -04 specification: 5’683 lines of text (last time was 6’574) Objects on encoding model changed 3 (last time was 7)
- Ideas Discussed and Scrapped: Dozens and dozens ;-)

What Changed ?

- Security Envelope and Security Model
- Packet Numbering (Debugging and Loss Detection)
- Sequence Number Arithmetic
- Link Capabilities
- More Tightening of Flooding Rules
- Clarifications on Flood Reduction Based on Open Source Implementation
 - Flooding In-Cast Reduction Explanation

Security Model

- Port-Association Model
- Node-Association Model
- Fabric-Association Model
- Can prevent all known attacks
 - Lifetime protected
 - Nonce exchange prevents replay
 - Origin integrity
- Does NOT
 - Provide confidentiality
 - Provide a chain-of-trust



Security Envelope

- RIFT Magic
- Outer Key Envelope
 - Generated link by link
 - Protected by Nonces, encapsulates TIE Lifetime
- Inner Key Envelope (only on TIEs)
 - Protects TIE
 - TIE can be passed through opaquely without deserialization
 - Allows to extend the model with optional elements without breaking backwards compatibility

```
UDP Header:
+++++
| Source Port | RIFT destination port |
+++++
| UDP Length | UDP Checksum |
+++++
| RIFT MAGIC | Packet Number |
+++++
```

```
Outer Security Envelope Header:
+++++
| Reserved | RIFT Major | Outer Key ID | Fingerprint |
| | Version | | Length |
+++++
|
~ Security Fingerprint covers all following content ~
|
+++++
| Nonce Local | Nonce Remote |
+++++
| Remaining TIE Lifetime (all 1s in case of LIE) |
+++++
```

```
TIE Origin Security Envelope Header:
+++++
| Inner Key ID | Fingerprint |
| | Length |
+++++
|
~ Security Fingerprint covers all following content ~
|
+++++
```

```
Serialized RIFT Model Object
+++++
|
~ Serialized RIFT Model Object ~
|
+++++
```

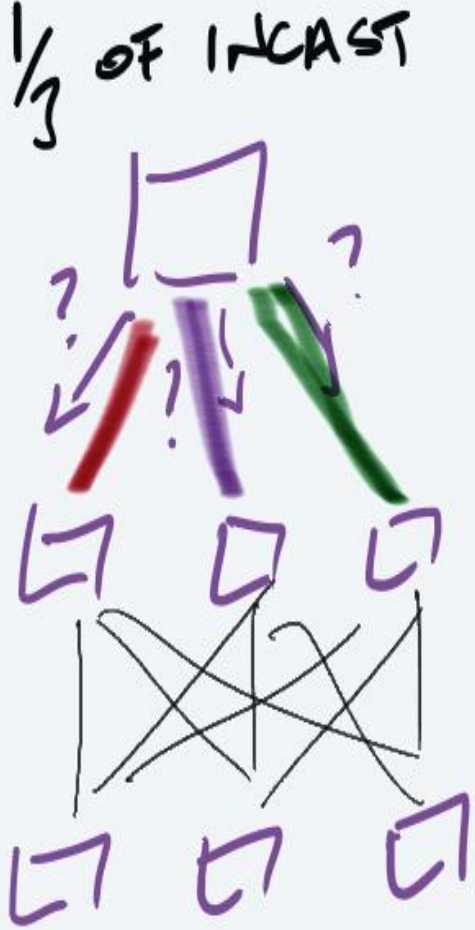
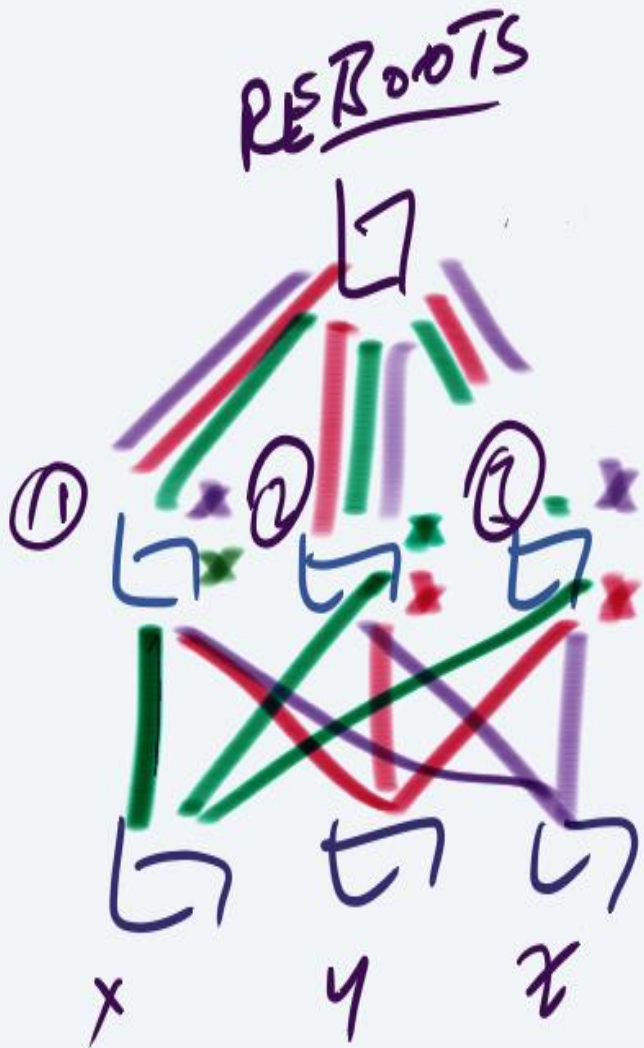
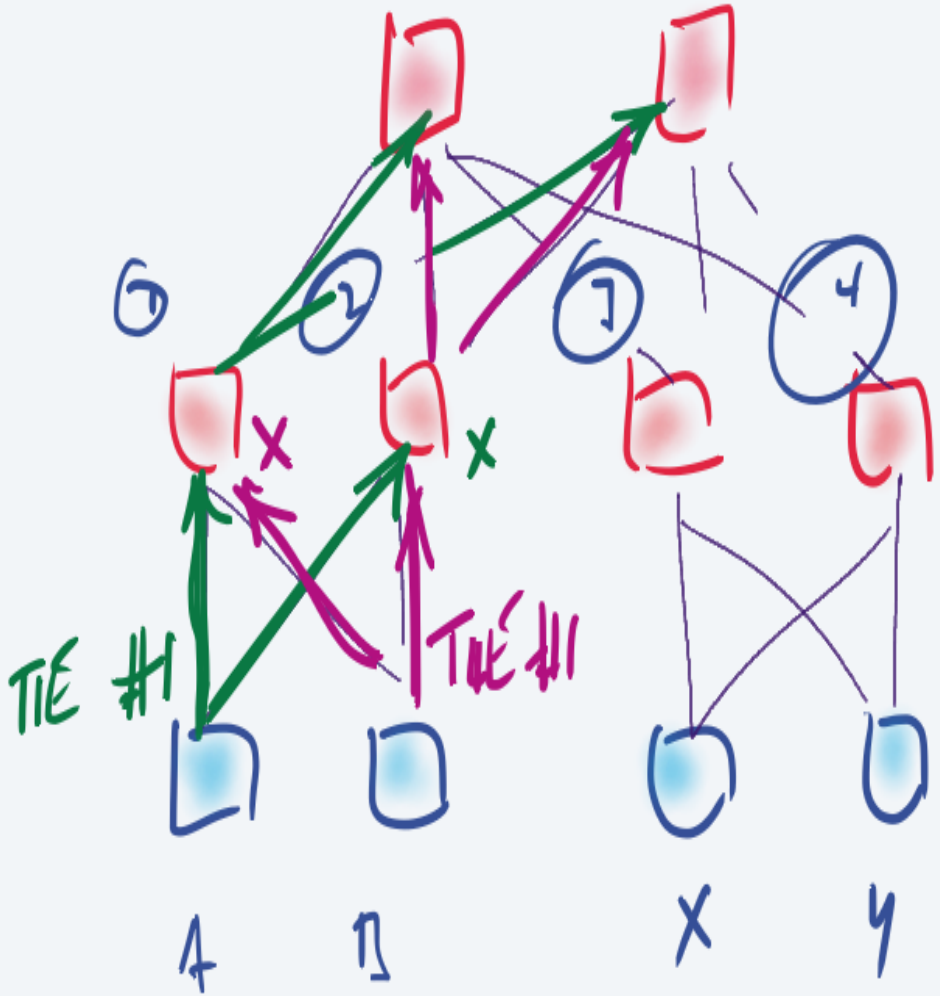
New Link Capabilites

- Link
 - BFD available or not

Final Cases and Tightening of Flooding Rules

- Open Source Implementors ask lots of questions ;-)
- Lifetime difference where lifetimes equal specified (60 secs)
- Last case of flooding inconsistency
 - Only occurs if 3 levels reboot in a specific sequence and the bottom generates a lower number on its N-TIE that the biggest held and middle cannot get
- Cosmetic change that asks a node to flush all other TIEs than its own on level change
 - Was leading to harmless but also useless TIEs hanging in a node until expiry

Flooding Incast Solved and Explained



Smaller Stuff

- Packet numbering on each packet type
 - Optionally increasing on each sent packet
 - Allows detection of losses on remote side and throttling
- Sequence Number Arithmetic
 - On TIE Types
 - Sequence Numbers
- Type Tightening
 - Types got smaller to save space (we don't be too concerned about it but wasting if we can be more frugal is not wise)
 - Seq#
 - Packet Numbers
 - Level Type
 - Version Types
- Unsolicited, optional downstream label

THANK YOU FOR YOUR ATTENTION