DC Routing protocol requirements

Jeff Tantsura

Dmitry Afanasiev

Keyur Patel

Petr Lapukhov

Tony Przygienda

Russ White

Yingzhen Qu

Jim Uttaro

Kenji Kumaki (post IETF 100)

Why DC napkin protocol design team?

Because we are long time friends ©



Why DC napkin protocol design team? Seriously

We know how to build routing protocols and DC's





Why DC routing protocol req's draft?



Why DC routing protocol req's draft?

Avoid protocol beauty contest - Have a single set of requirements to be

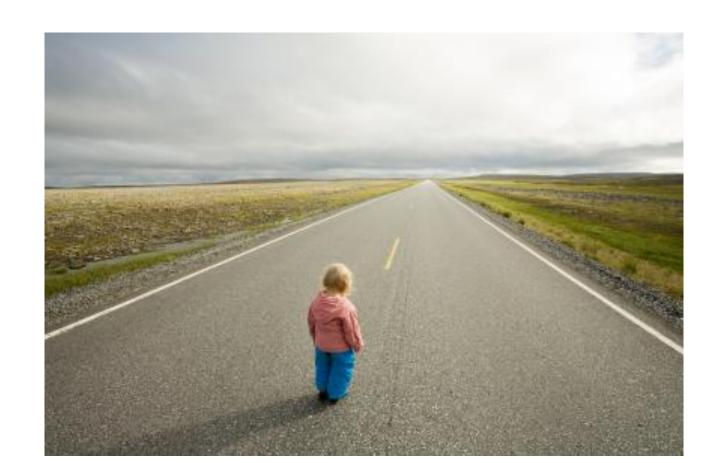
compared against





Why DC routing protocol req's draft?

We are just starting – we need your help!



Fabric definition

- The Fabric provides basic connectivity, with possibility to carry one or more overlay
- The Fabric provides no domain separation within, if needed, to be handled by an overlay
- Fabric' characteristics:
 - Regular (repeating identical or similar blocks) and usually recursive topology
 - Very high bisectional bandwidth and path diversity.
 Usually ECMP (anomaly of certain graph theoretical algorithms), some will require non-shortest paths and adaptive routing to be efficient.
 - Limited physical diameter (upto 80-100km) and small propagation delay with empty queues. One way propagation time < desired convergence time. Latency can be asymmetrical (longer physical links in some parts of fabric).

Number of links > number of nodes

15-Nov-17 7

Fabric req's

- The Fabric MUST support non equidistant end-points
- The Fabric MUST support Spine and Leaf + isomorphic topologies (Butterfly and similar)
- The Fabric MAY support non Spine and Leaf topologies
- The Fabric MAY provide interconnect facility for other fabrics

• ...

15-Nov-17 8

Fabric's KPI's: single-dimensional and expected to be changed...

- The Fabric SHOULD support 250k routes @ 5k fabric nodes with convergence time below 250ms
- The Fabric SHOULD support 500k routes @ 7.5k fabric nodes with convergence time below 500ms
- The Fabric SHOULD support 1M routes @ 10k fabric nodes with convergence time below 1s

 Combination of # of routes vs # of paths vs desired convergence time will be discussed in a later version

 The Fabric routing protocol MUST support load balancing using ECMP, wECMP and UCMP

 The Fabric routing protocol MAY support any custom or adaptive load balancing algorithms

- Fabric routing protocol SHOULD support route scale and convergence times of a Fabric mentioned above
- The Fabric routing protocol SHOULD support ECMP as wide as 512 paths
- The Fabric routing protocol MUST support various address families that covers IP as well as MPLS forwarding
- The Fabric routing protocol MUST support extensions to carry 3rd party data and Opaque data

Encoding and transport will be covered in a later version

- The Fabric routing protocol MUST support in band as well as out of band management
- The Fabric routing protocol MUST support Zero Touch Provisioning (ZTP)
- The Fabric routing protocol MUST support Neighbor Discovery to facilitate ZTP.
- The Fabric routing protocol MUST be able to leverage BFD for neighbor state(RFC5880)
- The Fabric routing protocol SHOULD be capable of bootstrapping a BFD session(RFC5882)

Fabric routing protocol operational requirements

- The Fabric routing protocol MUST be able to support real time state notifications of routes and its neighbors state to facilitate control plane telemetry
- The Fabric routing protocol MUST be able to support on-demand snapshots of protocol state and real time state notifications of routes and its neighbors state to remote node(s) to facilitate control plane telemetry
- The Fabric routing protocol MUST be able to handle commission/decommission of a node as well as any node restart with a minimal data plane impact

Fabric routing protocol requirements Following items have been identified to be studied at a later time:

- gRPC/THRIFT/similar encodings
- Ability to function as an overlay
- Flowlets signaling
- Multicast
- Auto aggregation/conditional de-aggregation
- State representation NB
- Integration with PCE/SDNc

QUESTIONS?

15-Nov-17 15

Next steps?