BIER
Bit Indexed Explicit Replication
Traffic Engineering
draft-eckert-bier-te-arch-02
IETF BIER-WG Yokohama 11/2015

Toerless Eckert, eckert@cisco.com
Gregory Cauchie, GCAUCHIE@bouygues TELECOM FR
BIER-TE -02 overview

• Enhanced documented based on discussion on list:

• Definitions, typos, references

• Refined description of saving BitPositions in rings
  One Bit to send from one pair of ingres routers to whole ring
  Two bits to send from any ingres router into the ring
    (clockwise, counter-clockwise)

• New section:
  How to deal with SI, subdomains and BFR-ID in BIER-TE
Definitions, BFR-ID in BIER-TE

• BIFT definition now aligned with BIER arch
  Every subdomain has a single BIFT. Index: SI:BitPosition.
    Eg: bitstring length 256, up to 256 SI == up to 64k indices in BIFT
    In BIER, every used SI:BitPosition (“bit”) is a BFER == BFR-ID

• What is a BFR-ID in BIER-TE?
  NONE: BIER-TE itself does not need/care about BFR-ID
    But BIER-encap, flow-overlay want to know BFR-ID

  In BIER-TE, every used SI:BitPosition indicates some adjacency
    BFER, group of BFER, transit adjacencies, ...
    No simple 1:1 mapping of bits to BFER (BFR-ID)
BFR-ID in BIER-TE

• Assigning BFR-ID:
  Each BFER will have one or more unique bits (SI:BitPosition).
  Controller picks lowest number of these as BFR-ID for BFER.

• Translate BFR-ID to bitstring:
  Independent branches: Each BFR-ID translates to set of bits
  Bitstring is OR of bitset for the desired BFER
  API: application/flow-overlay (on BFIR) gets (from “controller”) bits for every BFER a bit-set and can then independently calculate bitstrings by OR’ing

  Interdependent branches:
  Eg: steiner-tree: BFER is added/removed, desired TE-path to other BFER changes.
  Bitstring can only be determined by just OR’ing bits for each BFER.
  API: application/flow-overlay (on BFIR) indicates set of BFER and gets bitstring.

• Translate bitstring to list of BFR-ID:
  Equally possible, but not needed? (bits reset in forwarding, not seen by receiver).
Dealing with SI vs. Subdomains

• How does BIER use SI vs. Subdomains ?!
  – Authors best understanding…
  – Make BIER-TE behave the same (as much as possible)

• Subdomains for replication efficiency
  – Assume subdomain 0 is covering all BFER
    • Any service could just use this one subdomain.
  – Create additional subdomains for services requiring only subsets of BFER
    • To achieve more efficient replication than in subdomain 0:
    • Subdomain 0, packet to N BFER may require N copies if each BFER in different SI.

• Flow overlay behavior
  – SIs are transparent – flow overlay only works with BFR-ID and subdomain-IDs.
  – Each instance of a flow-overlay service needs to be configured with one or more subdomains it should use.

SI simple, subdomains relatively complex ->
A simple network
A simple network

Bits with SI = 1
Bits with SI = 2
Bits with SI = 3
Bits with SI = 4
Bits with SI = 5
Bits with SI = 6
A simple network

© 2015 Cisco and/or its affiliates. All rights reserved.
A simple network
BFR-ID assignments in BIER

• We would like BFR-ID to be just “identifier”

• If the network becomes large, it can be helpful to assign BFR-ID with location of BFER in mind
  To save bandwidth
  SI part of BFR-id becomes “locator”.

• Plan for the future
  Sequentially assigning BFR-ID for newly deployed BFR can decrease replication efficiency in the future.
What could TE help?

Traffic not load balanced as desired:
- few big flows
- alternative path not ECMP,...
- Eg:BFR1e1 to BFR1a shorter than to BFR1b

Same traffic across too many links (what more “steiner” like tree).
Simple setup

- Area 1: BFR1e1 to BFR1e200
- Area 2: BFR2e1 to BFR2e200
- Area 3: BFR3e1 to BFR3e200
- Area 4: BFR1a and BFR1b
- Area 5: BFR2a and BFR2b
- Area 6: BFR3a and BFR3b

Legend:
- Red: Bits with SI = 1
- Blue: Bits with SI = 2
- Green: Bits with SI = 3
- Yellow: Bits with SI = 4
- Pink: Bits with SI = 5
- Cyan: Bits with SI = 6
Simple Setup

Bits with SI = 1
Bits with SI = 2
Bits with SI = 3
Bits with SI = 4
Bits with SI = 5
Bits with SI = 6
Advanced setup

- Area 1: BFR1e1, ..., BFR1e200
- Area 2: BFR2e1, ..., BFR2e200
- Area 3: BFR3e1, ..., BFR3e200
- Area 4: BFR1e1, ..., BFR1e200
- Area 5: BFR2e1, ..., BFR2e200
- Area 6: BFR3e1, ..., BFR3e200

- Core:

- Bits with SI = 1
- Bits with SI = 2
- Bits with SI = 3
- Bits with SI = 4
- Bits with SI = 5
- Bits with SI = 6
Advanced setup,

BFR1e1 ... BFR1e200

area1

BFR1e1 ... BFR1e200

area2

BFR2e1 ... BFR2e200

area3

BFR3e1 ... BFR3e200

area4

BFR1e1 ... BFR1e200

area5

BFR2e1 ... BFR2e200

area6

BFR3e1 ... BFR3e200

- Bits with SI = 1
- Bits with SI = 2
- Bits with SI = 3
- Bits with SI = 4
- Bits with SI = 5
- Bits with SI = 6
Example setup of BitPositions with BIER-TE

• Assign BitPositions with SI as locator.
  Each “area” has one SI (with room for growth)
  Make area small enough for BFER AND transit links in area.

• Result:
  Can unicast from every BFIR to each receiver area with TE within each area.
  And engineer (in example) exactly which egress area edge-router is used.

• Advanced Setup:
  Assign BitPosition for every area on each edge-router
  Now we can engineer explicit paths via ingres-area-edge-router and egress-area edge-router via two routed adjacencies.
Managing BitPositions between BIER and BIER-TE

• Subdomain is either for BIER or BIER-TE.

• Design “areas” independently.

• TE: some percentage of bits required for transit.
  Depending on topology and TE desires: 20..80% of total bits in SI
  Many parallel paths – with need to engineer for them – many bits needed.

• Share subdomain for BIER and BIER-TE?
  But that would reduce replication efficiency for BIER.
  Depends also on forwarding plane features:
    How do we distinguish BIER from BIER-TE packet
    One label per (subdomain,SI) -> can only do BIER or TE per sub-domain.