

# draft-prz-lsr-hierarchical-snps-01

## Optimizing CSNP Load at Scale

Tony P & Li, HPE Juniper Networking



# What's New?

- Massive Readability Rewrite (with Perplexity helping a bit)
  - Idea of HSNP Level Abandoned Completely
  - Emulation of Refreshes/Updates over Long Periods on Very Large Networks to Understand Collision Probabilities and Durations
    - Introduction of 48bits HSNP Hash on Top of Fletchers of Fragments
    - Adding ISIS MTU Length to the Fragment Fletcher Improves Entropy Significantly
-

# Why No HSNP Level/Rank Anymore?

- Implementation Experience Shows That
  - Different Parts of LSDB Are Best Compressed At Different “Resolution”
  - Compression Varies over Time and per Peer
- Strategy in Choosing Node Ranges Very Implementation Dependent
  - Statistics over Hash Mismatches
  - Distinguishing Link Flaps from Node Reboots
  - Availability of Caches
  - Packing of Node Ranges to Optimize #CNSPs Advertised on Mismatch
- As Mentioned Previously, Sync’ing All Peers on “Same Ranges” Seems Natural But Is Not Feasible

# “Gradient Descent” Analogy & Resulting Advertisement Rules

- HSNPs Can Be Considered a “Fast, Steep Descent” Towards LSDB Consistency In Case Both Databases are Largely Synchronized
  - One Hash “Synchronizes” Much Larger Part of the Database Than a CSNP
  - As in Gradient Descent Going “Fast in the Wrong Direction” Means That Smaller Steps Need Be Taken to Retract/Refine the Direction Better
- Resulting Advertisement Rules Are Simple (Draft Needs to Formalize Those in Normative Language Ultimately):
  - On Reception of a Mismatched HSNP Hash over Range of Nodes the Receiver MUST Take Any of the Actions:
    - Send HSNP Packet With `_More Specific_` HSNP Hashes Covering The Range (i.e. Disaggregate)
    - Send CNSP/PSNP Packets Covering All Fragments of All the Nodes in the Mismatched Range
    - Flood All the Involved Fragments To the Sender
  - On Reception of a Mismatched HSNP **Node** Hash (i.e. Covering a Single Node ID Incl. Its PseudoNodes) the Receiver MUST Take Any of the Actions:
    - Send CNSP/PSNP Packets Covering Fragments of All the Fragments in the Mismatched Node ID Incl. Its PseudoNodes
    - Flood All the Fragments of the Node Incl. Its PseudoNodes

# Minor Changes and Considerations

- In Case of 48 bits Fragment Fletcher == 0 to Be Replaced with 1
- Case of Different Set of Fletchers Creating Same Node Hash is Too Unlikely to Consider
- The Case of Advertising Node Range Hash as 0 to Indicate Absence Needs Removing
  - Indistinguishable from a “Real” 0 Hash over a Node Range
  - Superfluous Since “Absence of the Range” in HSNP is Enough



# HSNP Fragment Hash Collision Simulation

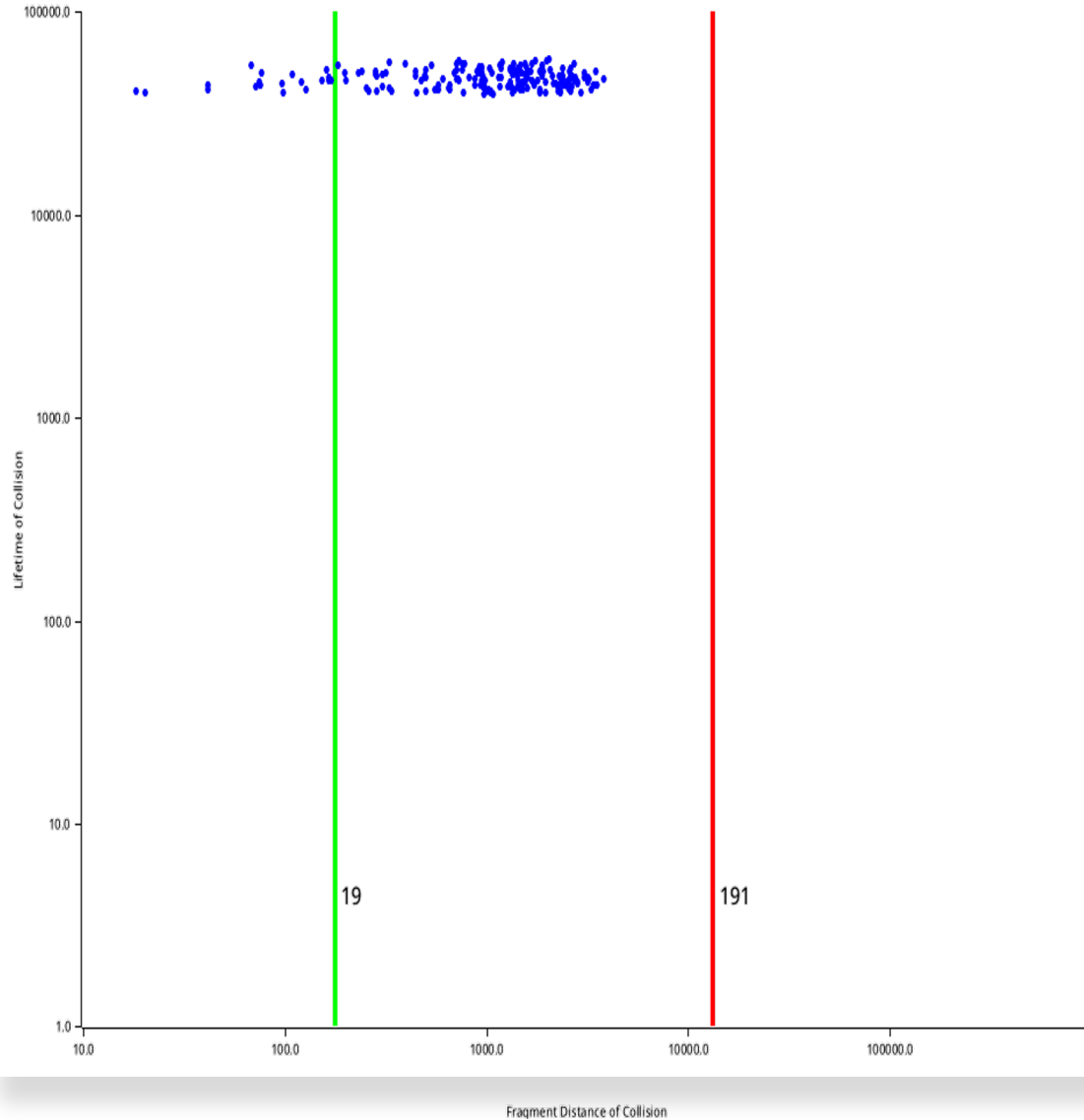
- Same HSNP Fragment Hash Within HSNP Node (Range) Hash Masks Out Colliding Fragment
  - Simulation
    - 32 Networks with 50K Nodes and ~1E6 Fragments
      - 2 Years Runtime per Network
    - Roughly Max Age Used To Refresh Fragments
    - Node IDs Differ by 3 Bytes Only
    - 5% of Refresh Changes ISIS PDU Length
    - ISIS Checksums Change More or Less in Uniform Fashion on Refreshes
    - Checks for Same HSNP Fragment Hash in Whole Database on Every Refresh
-

# Collision Probabilities

- 32 Bits Collides Too Much, Discarded Early
  - Comparison Between 48 and 64 Bits Hashes Best Shown as Graph
    - Horizontal is Distance Between Fragments That Collided in LSDB
    - Vertical is Duration of the Collision in Seconds
    - Red Line is # Fragments in Hash For Single HSNP Packet Covering Whole Database
    - Green Line is Roughly 80 HSNP Packets Covering Whole Database
    - Numbers Close to Lines Are Collisions Left of the Line
      - Only Relevant Collisions Since Both Fragments Are in the Same HSNP Hash
  - Less 64 Bits Hashes Fit Into HSNP Packet Hence They Cover Bigger Ranges
  - "Perfect" Remediation Techniques of Collisions Possible
    - Unlikely Practical Given Short Duration of Collisions and Large Distances in Database
-

# 64 Bits

48Bits hash 32 networks \* 50000 nodes over 2 years 952.17 ki fragments per network total refreshes/collisions 36.64 Gi/191.00



# 48 Bits

8Bits hash 32 networks \* 50000 nodes over 2 years 952.22 ki fragments per network total refreshes/collisions 36.64 Gi/236.00

