Moving Beyond End-to-End Path Information to Optimize CDN Performance, Krishnan et al.

Chris Moran

Motivation

- CDNs try to improve client experience by redirecting clients to the node with least latency
  - Done via prefix matching on client's DNS resolver
- But, clients still suffer from inflated latencies in this prefix block
  - Previous solution, add nodes or reposition nodes
  - What else can we do???
  - ~~~~ WhyHigh ~~~~

WhyHigh Goals/Questions

- Understand how well the latency based redirection is currently working
  - How well is current system performing? What is the best RTT we should see?
- Identify broad categories of causes for poor RTTs
  - Why are clients underperforming? Can we group clients together into categories?
- Detect poor RTTs and diagnose root causes
  - What can be done to alleviate the problems? What causes are we able to diagnose? Fix?

Non-goals

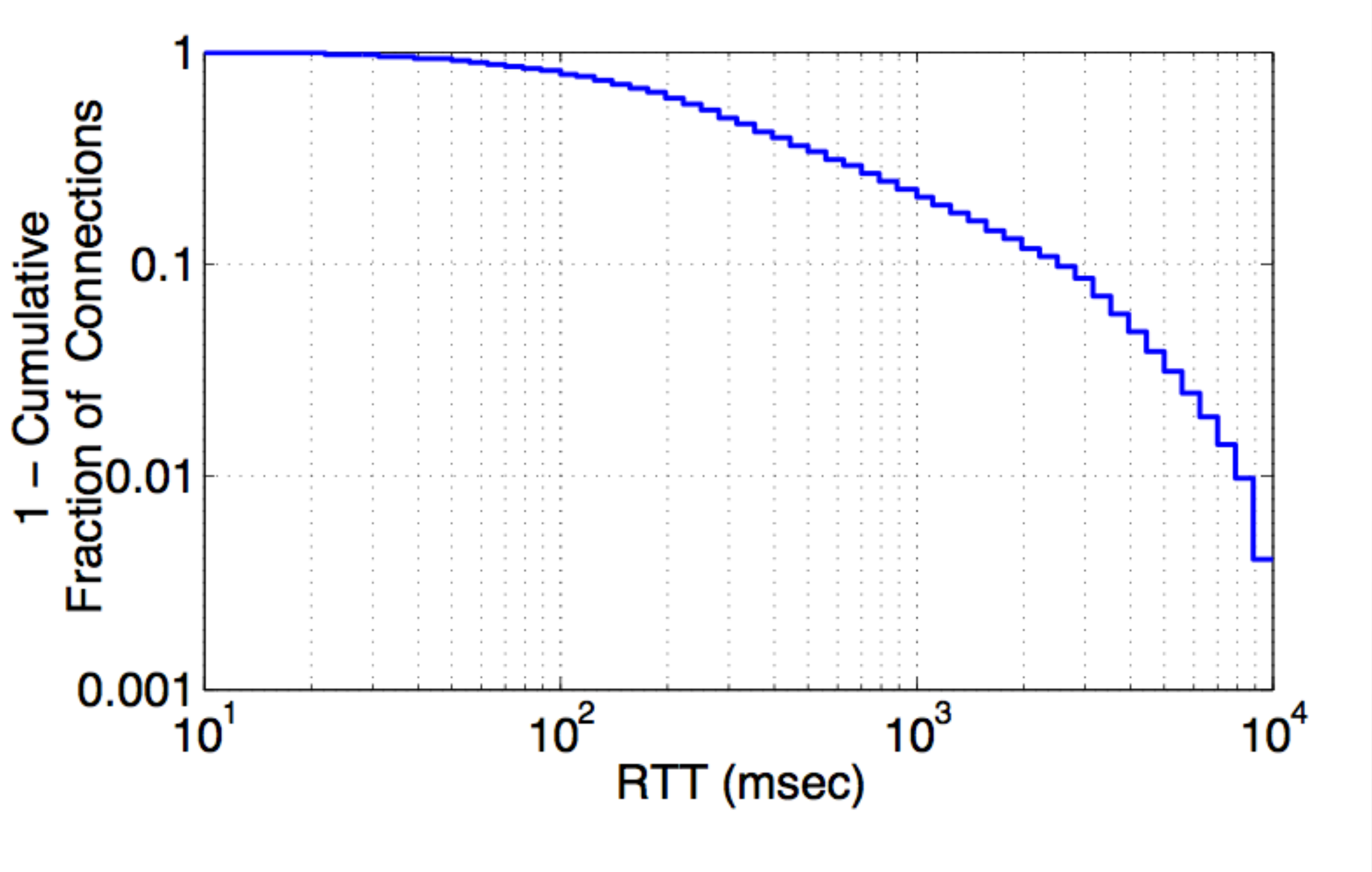
- Restrict the problem to end-to-end latencies
- Ignore cases where base assumptions of CDN are violated
  - DNS server co-location
  - Prefix block with wide-geographic distribution

- Focus where CDNs should be performing well, but are not.

Data Set and Pre-Processing

- Collect data from 13 Google CDNs, serve 170k prefixes
  - Log time between SYN-ACK and SYN-ACK-ACK, small packet size
  - Ignore cases where packet loss occurred
- Pre-processing
  - Min RTT < Geo-RTT
    - Wrong geo-location? Removes 21k
  - Inaccuracies in location based on confidence
    - Removes 42k
  - Geographically-wide prefixes
    - ~5k

Current RTT Distribution



What is the source of latency?

- Node serving not geographically closest?
  - Nope, 80% are directed to the closest node, 92% within 10ms of closest
- Routing Inefficiencies
  - Seen when minimum RTT for a block is significantly higher than region
- Queueing delays
  - Large variances in RTTs from clients

How to ID latency inflation?

- Look for inflated prefixes
  - (Min-RTT prefix) – (Min RTT region) > 50ms
- Figure out the cause
  - Forward path? CDN → client
    - Use traceroute to see routed path
  - Reverse path? CDN ← client
    - Traceroute, significant increase in single hop delay, circuitous path
    - Pings, drop in TTL in successive implies asymmetry
    - Flow records at Google network edge

Grouping Prefixes & Ranking Problems

- Techniques
  - Same PoP-level path
  - Same AS path, entry/exit point to Google network
  - Same AS path
  - Belong to same AS
- Limit number of cases
- Rank
  - Fraction of nearby prefixes w/ inflated latency
  - Fraction of nearby prefixes served elsewhere
  - Typically newer nodes have more issues

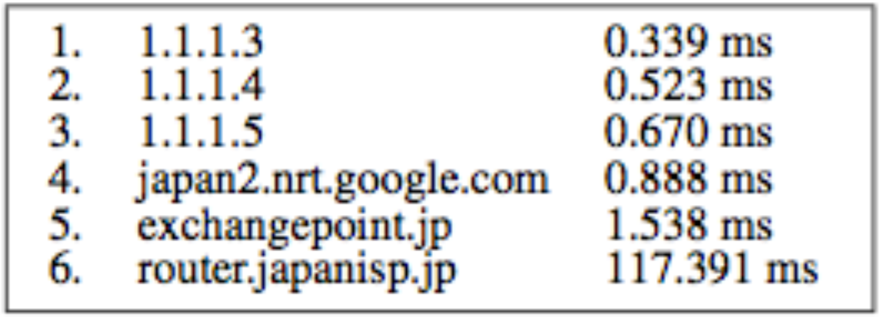
Causes of Inflation

- Lack of peering
  - AS paths are long
- Limited bandwidth capacity
  - Limited bandwidth between Google and AS
- Routing Misconfiguration
  - Reverse path is circuitous
- Traffic Engineering
  - Alternate shorter paths exist

WhyHigh in action, identifying

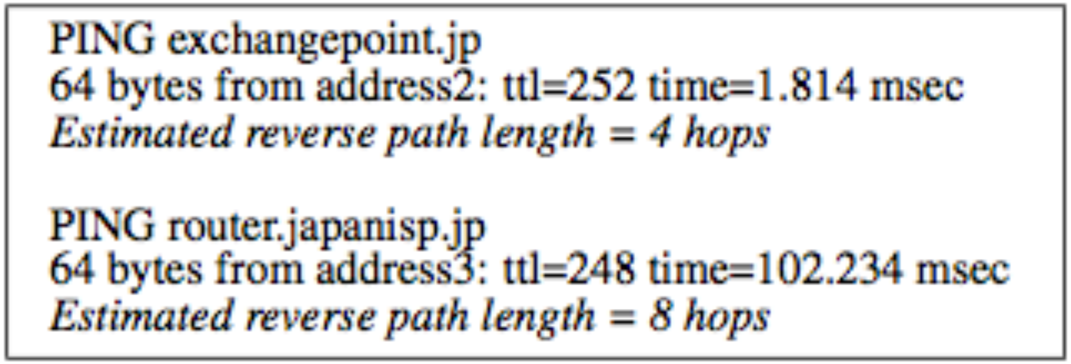
- PhilISP1
  - Southeast Asia ISP, neighbor AS(s) peering connecting to Google in US
- IndISP1
  - Lack of capacity between neighbor IndISP2, used longer route via ISP in Japan
- PhilISP2
  - Splitting address space, bad ISP policies
- JapanISP
  - Reverse path asymmetries due to legacy configuration of Google location

Example of inflated reverse path



```
1.   1.1.1.3                  0.339 ms
2.   1.1.1.4                  0.523 ms
3.   1.1.1.5                  0.670 ms
4.   japan2.nrt.google.com    0.888 ms
5.   exchangepoint.jp         1.538 ms
6.   router.japanisp.jp       117.391 ms
```

(a)



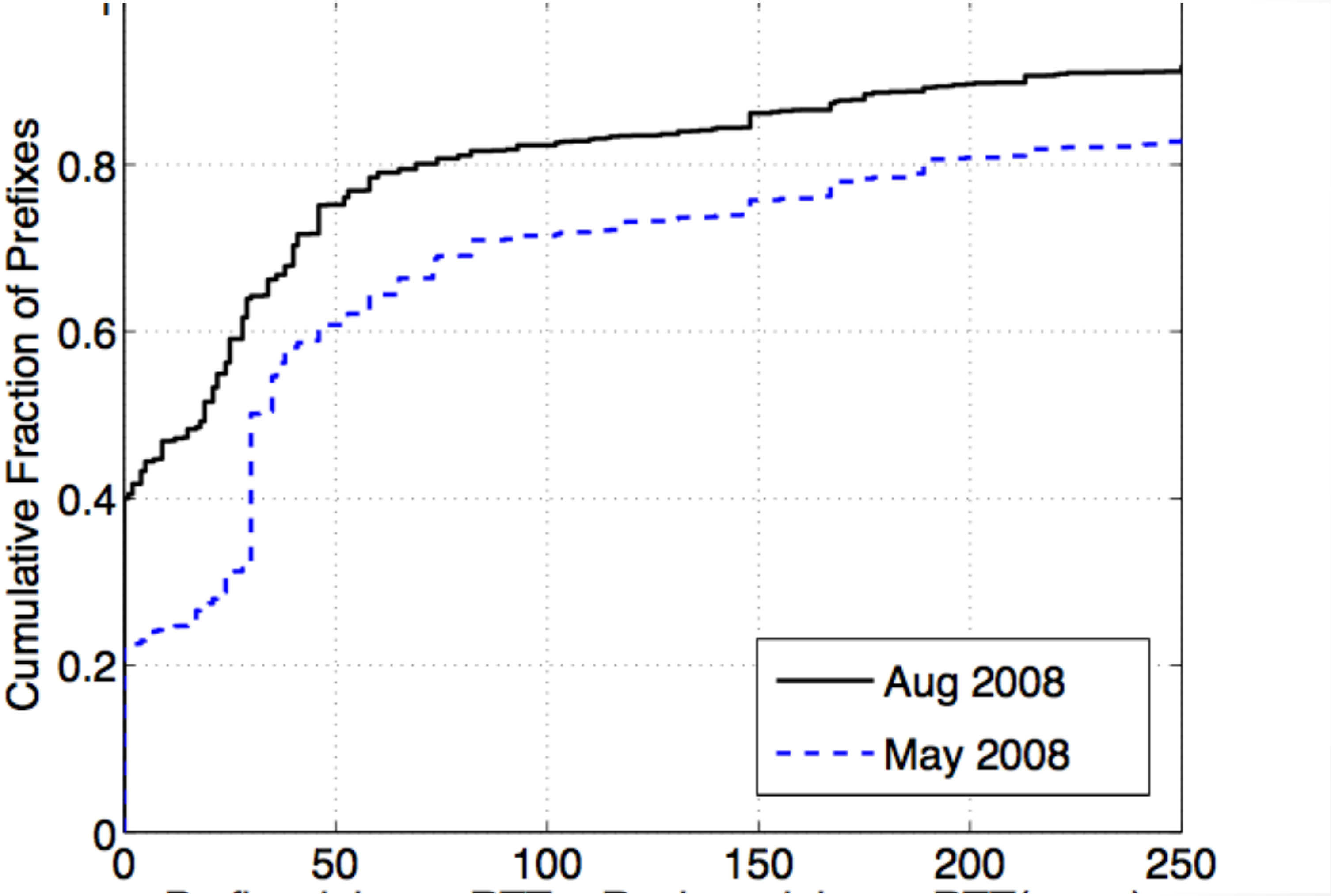
```
PING exchangepoint.jp
64 bytes from address2: ttl=252 time=1.814 msec
Estimated reverse path length = 4 hops

PING router.japanisp.jp
64 bytes from address3: ttl=248 time=102.234 msec
Estimated reverse path length = 8 hops
```

(b)

WhyHigh Fixes

- Resolve Japan ISP’s large latencies through a routing announcement
- PhilISP2 advertises a less specific prefix and utilizes Google’s peering link
- Still in process of fixes
  - 22% of ISPs w/ inflated paths use direct peering with Google

Performance gains, South America



Limitations of System

- Only partial view of the internet
  - Add measurement node
  - Reverse traceroute?
- Only see IP layer and above with traceroute
  - MPLS tunnel
- Can’t pinpoint queueing delays

Overall Impressions

- Very solid paper
- System tries to optimize the current configuration of a CDN while fixing possible problems with the network