Speeding Up CDNs with Subnet Assimilation from the Client

Authors:
Marc Anthony Warrior
Uri Klarman
Marcel Flores
Aleksandar Kuzmanovic
Northwestern University

CoNEXT ‘17
Incheon, South Korea
CDN & Caching Session
Bird’s Eye View

● What is Drongo?

● Why we need Drongo

● Performance Analysis

● Thoughts & Conclusions

● Questions
What is Drongo?
What is Drongo?

It’s a bird!
What is Drongo?

It’s a bird!
What is Drongo?

It’s a bird!
What is Drongo?

It’s a system that allows end-users to enhance the QoS (quality of service) they get from Content Distribution Networks (CDNs).
What is Drongo?

It’s a system that allows end-users to enhance the QoS (quality of service) they get from Content Distribution Networks (CDNs)

(in this talk, QoS = latency)
Why Latency?
Latency is **time**.
● Latency is **time**
● Latency is **money**
  ○ Google (Marissa Mayer), Amazon (Greg Linden)
    ■ Web 2.0 Summit, glinden.blogspot.com
● Latency is time

● Latency is money
  ○ Google (Marissa Mayer), Amazon (Greg Linden)
    ■ Web 2.0 Summet, glinden.blogspot.com

● Latency is the bottom line
  ○ “What we have found running our applications at Google is that latency is as important, or more important, for our applications than relative bandwidth,” Amin Vahdat (Google)
Drongo helps you *(the end user)* *lower your own latency!*
Drongo’s Effect on Latency

Google  Amazon  Alibaba  CDNetworks  ChinaNetCtr  CubeCDN

![Box plot showing latency ratio for different CDNs](image-url)
Drongo’s Effect on Latency

Google  Amazon  Alibaba  CDNetworks  ChinaNetCtr  CubeCDN

ONLY client-side changes
Example Scenario
Provider wants to serve client
Client is far
CDN = more replica locations
Which replica serves the client?
Choose the “closest” server
Choose the “closest” server

This choice is nontrivial!
Often Suboptimal Choices!
Maybe just a far LDNS...

[Chen - SigComm ’15; Huang - SigComm CCR ‘12;
Alzoubi - WWW ‘13; Rula - SigComm ‘14 …]
Ordinary DNS Query

DNS Query | LDNS IP

Somewhere in California
EDNS0 Client-Subnet extension (ECS)

DNS Query | LDNS IP | Client Subnet

Somewhere in California | Actually somewhere in New York
We used ECS:
We used ECS:

This still happens
We used ECS:

This still happens

... frequently
Really? ...
Really? ...

YES!
We measured it!
How did we measure it?
How did we measure it?

Find subnets directed to different replicas
## Subnet Assimilation

<table>
<thead>
<tr>
<th>DNS Query</th>
<th>LDNS IP</th>
<th>Client Subnet</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>DNS Query</th>
<th>LDNS IP</th>
<th>Other Subnet</th>
</tr>
</thead>
</table>

**Subnet Assimilation**
How did we measure it?

**search**
Find subnets directed to different replicas

**measure**
Perform pings and downloads to each replica
How did we measure it?

search
Find subnets directed to different replicas

measure
Perform pings and downloads to each replica

compare
Identify which subnet resulted in the “best” replica
1. Get “Default” Choice

(use client’s own subnet for ECS)
2. Traceroute to default choice
3. Get Hop Subnet Choices

(use hops’ subnets for ECS)
4. Measure Latencies
4. Measure Latencies

Steps 1-4: a “trial”
Latency Ratio

Normalize to default choice’s RTT

CLIENT

0.6
1
1.4
We’re looking for this
Valley = better choice from hop subnet

replica choice for subnet

traceroute

RTT: client to replica
Valley = better choice from hop subnet

replica choice for subnet

traceroute

RTT: client to replica

100 ms

0 ms
PlanetLab Sees Valleys!
PlanetLab Sees Valleys!
PlanetLab Sees Valleys!

- Google: 20.24%
- Amazon: 14.02%
- Alibaba: 33.68%
- CDNetworks: 15.61%
- ChinaNetCenter: 27.42%
- CubeCDN: 38.58%

Room for improvement!
5. Use best subnet for ECS
5. Use best subnet for ECS

Get best mapping!
Are Valleys Predictable?

- Trials are not “fast”
Are Valleys Predictable?

- Trials are not “fast”
- We want valleys “on the fly”
Are Valleys Predictable?

- Trials are not “fast”
- We want valleys “on the fly”
- We need to find valley-prone subnets
Testing Persistence

consecutive trials
Testing Persistence
Latency Ratio Difference Over Time

Latency Ratio = (hop replica RTT) / (default replica RTT)
Testing Persistence

Window A VS Window B
Testing Persistence

VS

Window A

Window C
Testing Persistence

Window A

15 hours

VS

Window C

15 hours
Latency Ratio Difference Over Time

Latency Ratio = (hop replica RTT) / (default replica RTT)
Latency Ratio Difference Over Time

Latency Ratio = (hop replica RTT) / (default replica RTT)
Latency Ratio Difference Over Time

Latency Ratio = (hop replica RTT) / (default replica RTT)
Latency Ratio Difference Over Time

Latency Ratio = (hop replica RTT) / (default replica RTT)

SURPRISE! The Internet is crazy!
Filter: at least one valley

Subnet A
\{0, 0, 0, 0, 0, 0, V, 0, 0, 0, 0, 0, 0, 0, V\}

Subnet B
\{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}

Subnet C
\{V, V, V, V, 0, 0, 0, 0, 0, V, V, V, 0, V\}
Filter: at least one valley

Subnet A: \( \{0,0,0,0,0,V,0,0,0,0,0,0,0,0,V\} \)

Subnet B: \( \{0,0,0,0,0,0,0,0,0,0,0,0,0,0\} \)

Subnet C: \( \{V,V,V,V,0,0,0,0,0,V,V,V,0,V\} \)
Filter: at least one valley

Latency Ratio = (hop replica RTT) / (default replica RTT)
Filter: at least one valley

Latency Ratio = (hop replica RTT) / (default replica RTT)

very flat
Filter: at least one valley

Latency Ratio = \( \frac{\text{hop replica RTT}}{\text{default replica RTT}} \)

- very flat
- Close to zero
Parameter Exploration
How deep are the valleys from useful subnets?

\[ V_{\text{thresh}} = \]
Latency Ratio

V_{thresh}

Replicas

A  B  C

0.6  1  0.9
Latency Ratio

V_{thresh} = 0.6

A: 0.6

B: 1

C: 0.9

Replicas
How often do valleys occur in useful subnets?

\[ V_{freq} = \]
\[ V_{\text{freq}} = \frac{2}{5} \]
Valley-Prone Subnet

\[ V_{freq}^{2/5} = \]

Valley-Prone Subnet
$V_{freq} = \frac{2}{5}$

Valley-Prone Subnet
\[ V_{freq} = \frac{2}{5} \]

NOT Valley-Prone Subnet
Overview of Drongo:

1. Collect training window
Overview of Drongo:

1. Collect training window

2. Count the # of sufficiently deep valleys
Overview of Drongo:

1. Collect training window

2. Count the # of sufficiently deep valleys

3. Apply subnet assimilation
   a. Training window is *already* complete
   b. Both parameters met
System Wide Performance
System Wide Performance

![Graph showing latency ratio vs valley threshold]

- **Latency Ratio**
  - Values range from 0.95 to 1.25

- **Valley Threshold**
  - Ranges from 0.1 to 1.0

- **Line**
  - $v_f \geq 0.2$

---

88
System Wide Performance

![Graph showing latency ratio vs. valley threshold]
System Wide Performance

![Graph showing latency ratio vs. valley threshold for different values of $v_f$.]

- $v_f \geq 0.2$
- $v_f \geq 0.4$

The graph illustrates the latency ratio across various valley thresholds, with different lines representing different threshold values. The graph suggests that for $v_f \geq 0.4$, the latency ratio is better compared to $v_f \geq 0.2$.
System Wide Performance

![Graph showing latency ratio vs valley threshold for different values of $v_f$. The graph indicates that for $v_f \geq 0.2$, the latency ratio increases with the valley threshold, and it is better compared to other values.](image)
System Wide Performance

![Graph showing latency ratio vs. valley threshold with different thresholds marked by symbols and lines.

- $v_f \geq 0.2$
- $v_f \geq 0.4$
- $v_f \geq 0.6$
- $v_f \geq 0.8$]
System Wide Performance

![Graph showing system wide performance with latency ratio on the y-axis and valley threshold on the x-axis. Different symbols represent different threshold values. The graph shows that as the valley threshold increases, the latency ratio also increases, indicating worse performance.]
System Wide Performance

\[ V_{freq} = 1.0 \]
System Wide Performance

\[ V_{\text{freq}} = 1.0 \]

\[ V_{\text{thresh}} = 0.95 \]
## Switch Quality

<table>
<thead>
<tr>
<th></th>
<th>Google</th>
<th>Amazon</th>
<th>Alibaba</th>
<th>CDNetworks</th>
<th>ChinaNetCtr</th>
<th>CubeCDN</th>
</tr>
</thead>
</table>

**Global Params**

**Per Prov. Params**
Conclusion & Insights

- CDNs have a lot of room for improvement
Conclusion & Insights

- CDNs have a lot of room for improvement
- Clients can help
● CDNs have a lot of room for improvement
● Clients can help
● Low requirements
Conclusion & Insights

- CDNs have a lot of room for improvement
- Clients can help
- Low requirements
- Can provide 50% improvement
Questions?
# Clients Affected

![Graph showing the fraction of clients affected versus valley threshold. The graph includes lines and markers for different thresholds, e.g., $v_t \geq 0.2$, $v_t \geq 0.4$, etc.](image)
Performance of Drongo’s choices