How LinkedIn used TCP Anycast to make the site faster

Ritesh Maheshwari

Shawn Zandi
Anycast

- Anycast provides a distributed service via routing.
- It is not really different than unicast.
  - NLRI object with multiple next-hops.
- It simply works for both TCP and UDP applications. (use with cautions!)
Anycast with ECMP

• Not a real issue in today’s internet
• Consistent flow routing is required (per packet load balancing breaks Anycast) – Pretty Much Standard
• Most BGP implementations do not load balance across different AS-PATHs even with same size.
Anycast Complications

- Broken MTU Challenges
  - ICMP message may not reach the intended receiver to report MTU problem. Adjusting MSS can help.

- RPF Checks

- Multiple covering prefixes - Only one Service Address should be covered by each advertised prefix /24 or /56

- Monitoring!
But!
How to measure Anycast effectiveness?
What is RUM?

JavaScript (Client-code) to measure performance

- DNS Time
- Connection time
- First Byte Time
- Download Time
- Page Load Time
What are PoPs?

Point of Presence / PoP

• Small-scale data centers
• Proxy servers at LinkedIn (ATS)
Without PoPs

Browser → Data Center

connection time: 250ms
Without PoPs

Browser

Data Center

connection time

250ms

server compute time

500ms
Without PoPs

Browser

- Connection time
- First byte time + page download time

Data Center

- Server compute time
- 3-5 round trips

5 RTTs = 5x250ms = 1250ms

Total = 2000ms
With PoPs

Browser 100ms PoP 250ms Data Center
With PoPs

Browser  PoP  Data Center

connection time

100ms

Old TCP Connection
With PoPs

connection time

first byte time + page download time

100ms

Old TCP Connection

server compute time

one round trip

500ms
With PoPs

**900 ms gain!**

**Total = 1100ms**

- **Browser**
  - connection time
  - first byte time + page download time

- **PoP**
  - connection time: 100ms
  - 5 RTTs: 5x100ms = 500ms

- **Data Center**
  - Old TCP Connection: 500ms
  - server compute time
  - one round trip

**Connection time**

- **Browser** to **PoP**
- **PoP** to **Data Center**

**Total time**

- 1100ms

**Gain**

- 900ms
How are users assigned to PoPs?

Through DNS:
  IP handed based on user’s resolver country

# California
$ dig +short www.linkedin.com
216.52.242.80

# Spain
$ dig @109.69.8.51 +short www.linkedin.com
91.225.248.80
Should India connect to Singapore or Dublin?

How to assure optimal PoPs assignment?
RUM beacons

Fetch a tiny object from each candidate PoP

For each pop_name,
1. Start timer
2. Fetch \{pop_name\}.perf.linkedin.com/pop/admin
3. Stop timer
Send data back to our servers

• Millions of agents!
• Analyze data to find “optimal” PoP per country
We can assign countries to new PoPs!

<table>
<thead>
<tr>
<th>Country</th>
<th>PoP</th>
<th>Median Beacon Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>Hong Kong</td>
<td>434</td>
</tr>
<tr>
<td>China</td>
<td>Dublin</td>
<td>1216</td>
</tr>
<tr>
<td>China</td>
<td>Singapore</td>
<td>515</td>
</tr>
<tr>
<td>India</td>
<td>Hong Kong</td>
<td>1368</td>
</tr>
<tr>
<td>India</td>
<td>Dublin</td>
<td>1042</td>
</tr>
<tr>
<td>India</td>
<td>Singapore</td>
<td>898</td>
</tr>
</tbody>
</table>
We can audit current assignment!

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>India</td>
<td>TRUE</td>
<td>Singapore</td>
<td>Singapore</td>
</tr>
<tr>
<td>Pakistan</td>
<td>FALSE</td>
<td>Singapore</td>
<td>Dublin</td>
</tr>
<tr>
<td>Spain</td>
<td>TRUE</td>
<td>Dublin</td>
<td>Dublin</td>
</tr>
<tr>
<td>Brazil</td>
<td>FALSE</td>
<td>US West Coast</td>
<td>US East Coast</td>
</tr>
<tr>
<td>Netherlands</td>
<td>TRUE</td>
<td>Dublin</td>
<td>Dublin</td>
</tr>
<tr>
<td>UAE</td>
<td>FALSE</td>
<td>US West Coast</td>
<td>Dublin</td>
</tr>
<tr>
<td>Italy</td>
<td>TRUE</td>
<td>Dublin</td>
<td>Dublin</td>
</tr>
<tr>
<td>Mexico</td>
<td>TRUE</td>
<td>US West Coast</td>
<td>US West Coast</td>
</tr>
<tr>
<td>Russia</td>
<td>FALSE</td>
<td>US West Coast</td>
<td>Dublin</td>
</tr>
</tbody>
</table>
LinkedIn Homepage Download Time Improvement

<table>
<thead>
<tr>
<th>Country</th>
<th>Median Improvement</th>
<th>90th Percentile Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>India</td>
<td>25%</td>
<td>30%</td>
</tr>
<tr>
<td>Pakistan</td>
<td>20%</td>
<td>25%</td>
</tr>
<tr>
<td>Singapore</td>
<td>15%</td>
<td>20%</td>
</tr>
<tr>
<td>Russia</td>
<td>10%</td>
<td>15%</td>
</tr>
<tr>
<td>Brazil</td>
<td>5%</td>
<td>10%</td>
</tr>
</tbody>
</table>

*Countries are listed in alphabetical order.*
Success.
Plot Twist:
Assignment far from optimal

• About 31% of US traffic gets assigned to a suboptimal PoP.
  – 45% of East Coast

• About 10% of traffic globally gets assigned to a suboptimal PoP.
DNS PoP assignment is suboptimal

• Assignment based on Resolver IP, not Client IP
DNS PoP assignment is suboptimal

• Assignment based on Resolver IP, not Client IP

• Bad IP to Geo databases
  – Resolver really in NY, but database says CA
Story so far

1. We built PoPs
2. ...used RUM to assign users to Optimal PoPs
3. ...found DNS based assignment is suboptimal
Accurate PoP assignment Problem

• Bug our DNS providers (31% -> 27%)
• Run our own DNS

How about *Anycast*?
Anycast – One IP, Multiple Servers

✓ Client IP, not Resolver IP used!
✓ No Geo-IP Databases
How does Anycast compare to DNS?

Will anycast send more users to optimal PoP?

➤ Lets test it!
RUM to rescue

For each PoP:

1. Announce same anycast IP (108.174.13.10)
2. Configure a domain
   ac.perf.linkedin.com to point to 108.174.13.10
RUM to rescue

For each page view:
1. RUM downloads a tiny object:
   \texttt{ac.perf.linkedin.com/pop/admin}
2. Read \texttt{X-Li-Pop} response header to record which PoP served the object
3. Send this back to LinkedIn with RUM data

Data:
1. For each user, the anycast PoP
2. For each user, the optimal PoP (from pop beacons)
## Results 😊

<table>
<thead>
<tr>
<th>Region or Country</th>
<th>DNS % Optimal Assignment</th>
<th>Anycast % Optimal Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illinois</td>
<td>70</td>
<td>90</td>
</tr>
<tr>
<td>Florida</td>
<td>73</td>
<td>95</td>
</tr>
<tr>
<td>Georgia</td>
<td>75</td>
<td>93</td>
</tr>
<tr>
<td>Pennsylvania</td>
<td>85</td>
<td>95</td>
</tr>
</tbody>
</table>
## Results 😞

<table>
<thead>
<tr>
<th>Region or Country</th>
<th>DNS % Optimal Assignment</th>
<th>Anycast % Optimal Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arizona</td>
<td>60</td>
<td>39</td>
</tr>
<tr>
<td>Brazil</td>
<td>88</td>
<td>33</td>
</tr>
<tr>
<td>New York</td>
<td>77</td>
<td>74</td>
</tr>
</tbody>
</table>
Fewer hops != Lower Latency

- Carriers prefer to haul packets within their own network
- Peering can create inter-continental short cuts
Maybe DNS wasn’t so bad

Continent-level assignments

City / State level assignments
“Regional” Anycast

DNS-based
1 anycast IP per continent

Ran a RUM experiment, all was fine

Alice

inter-continental link

2.2.2.2

1.1.1.1

1.1.1.1

Y

Z
USA Ramp Results

Ramp outside USA
In progress
Story so far

1. We built PoPs
2. ...used RUM to assign users to Optimal PoPs
3. ...found DNS based assignment is suboptimal
4. ...evaluated Anycast as a solution using RUM
5. ...now using Anycast to assign users to PoPs

Next play:
• Build more PoPs!
Story: The End

Learnings
• Clients are your measurement agents
• Trust, but verify
• You can have a bigger impact if you collaborate

Next Play
• Keep evaluating Anycast
• Keep building new PoPs