This is an **Informational** draft

**Today:** first time presented at DNSOP

Versions and mailing list discussion:

- **-03 (2019-03-11):** (minor changes from -02)
- **-02 (2019-03-08):** link list thread (no responses)
- **-01 (2018-12-20):** link list thread (no responses)
- **-00 (2018-11-28):** link list thread

Github link:

- [https://github.com/gmmoura/draft-moura-dnsop-authoritative-recommendations](https://github.com/gmmoura/draft-moura-dnsop-authoritative-recommendations)

DNSOP chairs asked us to contact DNS OP folks to review

- [https://lists.dns-oarc.net/pipermail/dns-operations/2019-February/018411.html](https://lists.dns-oarc.net/pipermail/dns-operations/2019-February/018411.html)

- Got some good reviews, issues opened on GitHub
• 13 people that have had 5 research papers:
  • Draft authors + Ricardo de O Schmidt, Wouter B. de Vries, Moritz Müller, Lan Wei, Cristian Hesselman, Jan Harm Kuipers, Pieter-Tjerk de Boer and Aiko Pras.

• Relevant papers with *recommendations* backed by large-scale, Internet-wide measurements:
  • 4x ACM IMC
  • 1x PAM

• However, papers tend to be *long, detailed* – they explain *why*
This draft:

definitions = [ ]
definitions.append(Moura16b)
definitions.append(Mueller17b)
definitions.append(Schmidt17a)
definitions.append(Vries17b)
definitions.append(Moura18b)

for p in definitions:
    recommendations = TLDR(p)  # great filter :-) 
    print(recommendations)

- Tangible, direct recommendations to OP folks on what to do
- With references to papers to know why
- **Target group:** large authoritative DNS ops, with global traffic
Recommendations in a nutshell

- R1: Use equally strong IP anycast in every authoritative server to achieve even load distribution [1]
- R4: When under stress, employ two strategies [4]
- R5: Consider longer time-to-live values whenever possible [5]
- R6: Shared Infrastructure Risks Collateral Damage During Attacks [4]
R1: Use equally strong IP anycast in every authoritative server to achieve even load distribution

- **Resolvers** will query **ALL** authoritatives (NS) [1]
  - (conclusions drawn from Ripe Atlas, .nl and the Roots data)
- However, nearby authoritatives will receive more queries
R1: Use equally strong IP anycast in every authoritative server to achieve even load distribution

- For OPs: latency of all Auth servers matter
- Unicast cannot deliver good global performance
- [1] recommends: use anycast at all Auth servers
  - equally strong in peering and capacity; and **phase out** unicast.
- This recommendation has been deployed at .nl.
When evaluating an anycast DNS provider, people always ask: “how many sites/instances” do you have?

Assumption: more instances → lower latency

[2] shows that this is not always true:

- c-root: 8 instances.
- k-root: 33 instances
- l-root: 144 instances
- Their median RTT: 30–32 ms to 7.9k Atlas probes

In other words, similar latency values for different deployments
R2: Routing Can Matter More Than Locations

- Why? BGP is agnostic to geographical distance
  - A client in California may be answered by an instance near NRT
    - even though there is a closer instance in SFO
- [2] thus recommends carefully considering **routing** and **connectivity** when engineering DNS anycast services
  - 12 sites is enough to provide good global latency
  - However, more instances may be helpful in case of DDoS [4]
R3: Collecting Detailed Anycast Catchment Maps Ahead of Actual Deployment Can Improve Engineering Designs

- Say you run an anycast service with $n$ instances
- Say you want to add 1 more instance in LAX
- How will that affect traffic among your other locations?
  - **Very hard** to predict without measurement
Solution:
- measure anycast catchments *ahead* of deployment
- create anycast maps from these measurements

[3] presents an ICMP-based tool (Verfploeter) for this solution
- [https://github.com/Woutifier/verfploeter](https://github.com/Woutifier/verfploeter)

Applied to *b-root* to **predict** query load on LAX:
- Predicted: **81.6%**
- Actual: **81.4%**.

Current deployments:
1. Anycast testbed ([http://anycast-testbed.nl](http://anycast-testbed.nl))
2. B-root
3. Large unnamed operator
R4: When under stress, employ two strategies

- BGP will map traffic to locations
- Best course of action?
  1. **Do nothing and let LAX become a degraded absorber**
  2. **Withdraw/prepend routes to shift traffic**
- Best option depends on attack and NS specifics
R4: When under stress, employ two strategies

- BGP will map traffic to locations
- Best course of action?
  1. **Do nothing and let LAX become a degraded absorber**
  2. **Withdraw/prepend routes to shift traffic**
- Best option depends on attack and NS specifics

**DDoS Load distrib**

- LAX: 50%
- AMS: 20%
- NRT: 10%
- GRU: 20%

Your Anycast NS

- DDoS
R4: When under stress, employ two strategies

DDoS Load distrib:
- LAX: 50%
- AMS: 20%
- NRT: 10%
- GRU: 20%

Your Anycast NS

- BGP will map traffic to locations
- Best course of action?
  1. Do nothing and let LAX become a degraded absorber
  2. Withdraw/prepend routes to shift traffic

Best option depends on attack and NS specifics
R4: When under stress, employ two strategies

- BGP will map traffic to locations
- Best course of action?
  1. Do nothing and let LAX become a degraded absorber
  2. Withdraw/prepend routes to shift traffic
- Best option depends on attack and NS specifics
R5: Consider longer TTL values whenever possible

- TTLs set how long queries should remain in resolver’s cache
  - Sort of “ephemeral replication”
- [5] emulates DDoS attacks (50-100% packet loss)

Figure 1: TTL: 1h; 100% Packet loss after $t = 10\text{min}$
R5: Consider longer TTL values whenever possible

- Caching is a key component of DNS resilience
- Resolver’s retries as well
  - they may even “hammer” authoritative servers
- As such, [5] recommend longer TTLs whenever possible
  - There’s no one-size-fits-all solution here
R6: Shared Infrastructure Risks Collateral Damage During Attacks

- Be careful when engineering DNS services:
  - co-location implies you share (parts of the) infrastructure
- [4] found that when the Root DNS was attacked, some .nl co-located instances also suffered
- Dyn 2016 Attack shows similar results
  - multiple zones were only partially reachable when NSes were attacked
- Conclusion: be aware of shared infrastructure risk
Questions?

- R1: Use equally strong IP anycast in every authoritative server to achieve even load distribution [1]
- R4: When under stress, employ two strategies [4]
- R5: Consider longer time-to-live values whenever possible [5]
- R6: Shared Infrastructure Risks Collateral Damage During Attacks [4]

Thanks reviewers of draft versions

https://github.com/gmmoura/draft-moura-dnsop-authoritative-recommendations
