When the Dike Breaks: Dissecting DNS Defenses During DDoS

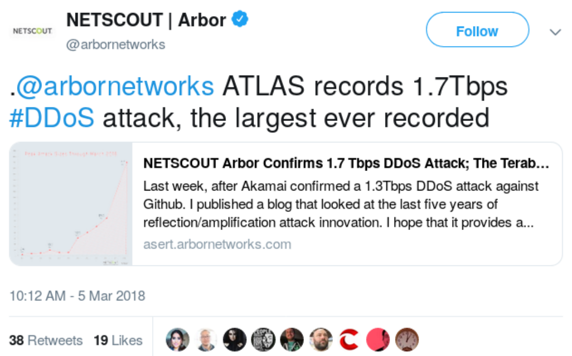
Giovane C. M. Moura[1,2], John Heidemann[3], Moritz Müller[1,4],
Ricardo de O. Schmidt[5], Marco Davids[1]

ACM IMC 2018, Boston, Massachusetts
2018-10-30

[1]SIDN Labs, [2]TU Delft, [3]USC/ISI,
[4]University of Twente, [5]University of Passo Fundo

DDoS Attacks

- DDoS attacks are on the rise [2, 1, 5]
- Getting bigger, more frequent, cheaper, and easier



NETSCOUT | Arbor
@arbornetworks

.@arbornetworks ATLAS records 1.7Tbps #DDoS attack, the largest ever recorded

NETSCOUT Arbor Confirms 1.7 Tbps DDoS Attack; The Terab...

Last week, after Akamai confirmed a 1.3Tbps DDoS attack against Github. I published a blog that looked at the last five years of reflection/amplification attack innovation. I hope that it provides a...

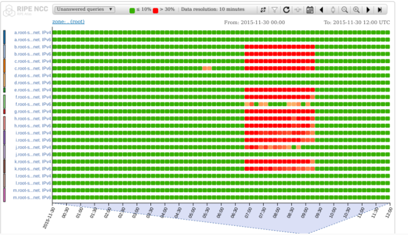
asert.arbornetworks.com

10:12 AM - 5 Mar 2018

38 Retweets 19 Likes

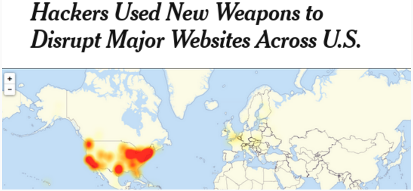
DDoS against DNS services

Root DNS DDoS Nov 2015



- **red** shows some sites were out, but no know errors
- **users**: **no known reports** of errors [3]

Dyn Oct 2016



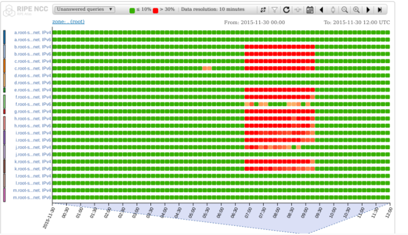
Hackers Used New Weapons to Disrupt Major Websites Across U.S.

- **users**: some users **could not reach** popular sites [5]: Twitter, Netflix, Paypal...
- even though Web servers were fine

Two large DDoSes, very different outcomes. Why?

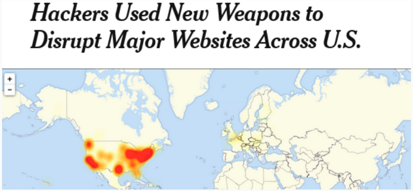
DDoS against DNS services

Root DNS DDoS Nov 2015



- **red** shows some sites were out, but no know errors
- **users**: **no known reports** of errors [3]

Dyn Oct 2016

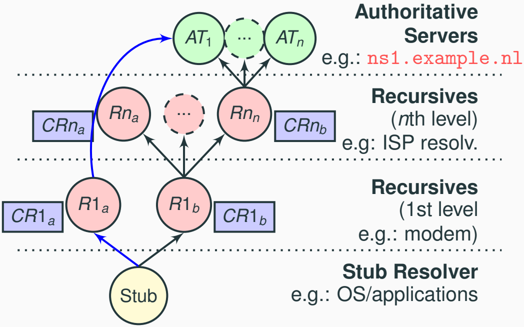


Hackers Used New Weapons to Disrupt Major Websites Across U.S.

- **users**: some users **could not reach** popular sites [5]: Twitter, Netflix, Paypal...
- even though Web servers were fine

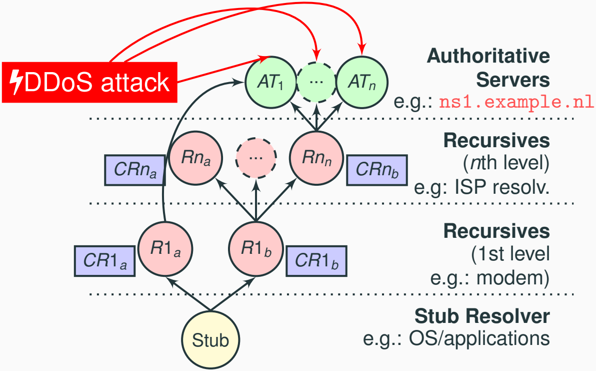
Two large DDoSes, very different outcomes. Why?

Background: the many parts of DNS



- Clients (stub) use recursives to resolve domains
- Recursives vary in **complexity and architecture**
- Authoritative servers answer with a **TTL value**: max limit to cache (CRn)

How are users affected by DDoS?



Authoritative Servers
e.g.: ns1.example.nl

Recursives
(*n*th level)
e.g: ISP resolv.

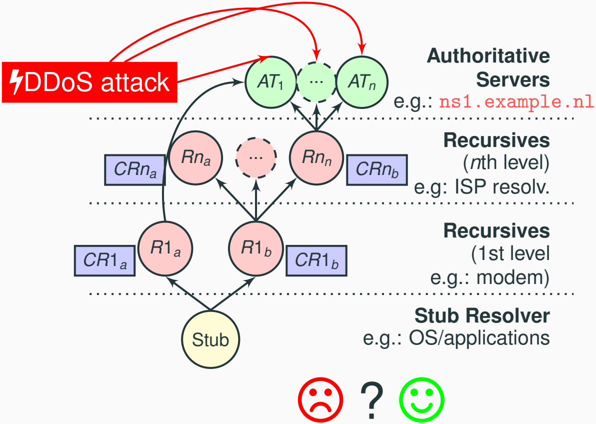
Recursives
(1st level
e.g.: modem)

Stub Resolver
e.g.: OS/applications

☹ ? ☺

- How much recursives's built-in defenses help user's experience?

How are users affected by DDoS?



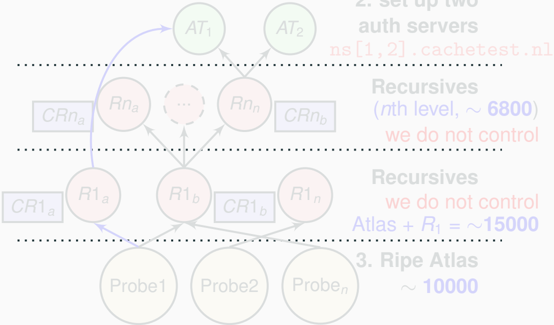
- How much recursives's built-in defenses help user's experience?

Evaluating DNS Resiliency

- **Part 1**: (a) define user experience and (b) evaluate caching
- **Part 2**: verify results of Part 1 in production zones (`.nl`)
- **Part 3**: emulate DDoSes in the wild to **to observe user experience**

Part 1: measuring caching in the wild

1. register `cachestest.nl`



2. set up two auth servers
`ns[1,2].cachetest.nl`

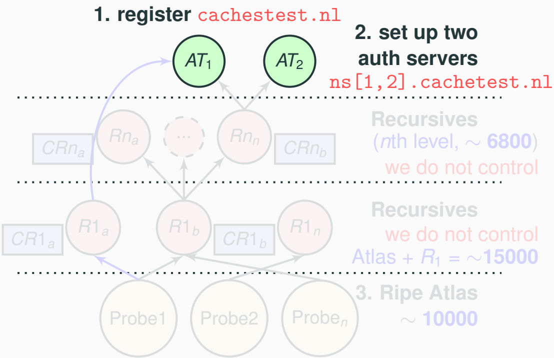
Recursives
(nth level, $\sim$ **6800**)
we do not control

Recursives
we do not control
Atlas + R_1 = $\sim$**15000**

3. Ripe Atlas
$\sim$ **10000**

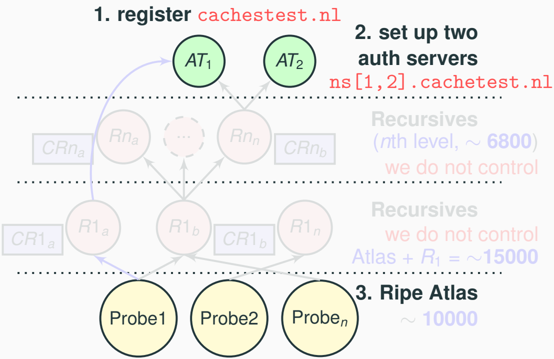
- Probes send unique queries to avoid cache interference
- Custom answers to tell if from cache or not (see Sec. 3.2)
- Probe every 20min, for 2 to 3 hours
- Various TTLs: 60, 1800, 3600, and 86400s
- 15000 Vantage Points, 6800 R_n (no DDos)

Part 1: measuring caching in the wild



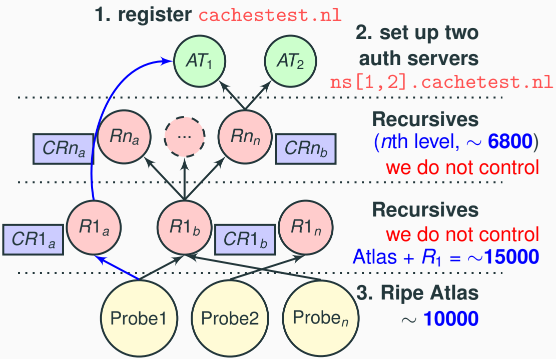
- Probes send unique queries to avoid cache interference
- Custom answers to tell if from cache or not (see Sec. 3.2)
- Probe every 20min, for 2 to 3 hours
- Various TTLs: 60, 1800, 3600, and 86400s
- 15000 Vantage Points, 6800 R_n (no DDos)

Part 1: measuring caching in the wild



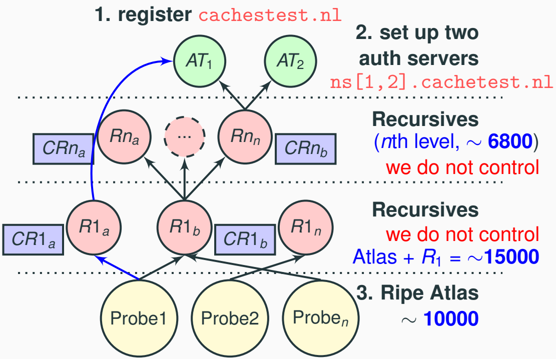
- Probes send unique queries to avoid cache interference
- Custom answers to tell if from cache or not (see Sec. 3.2)
- Probe every 20min, for 2 to 3 hours
- Various TTLs: 60, 1800, 3600, and 86400s
- 15000 Vantage Points, 6800 R_n (no DDos)

Part 1: measuring caching in the wild



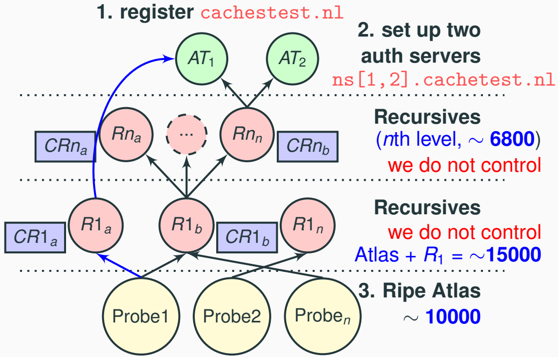
- Probes send unique queries to avoid cache interference
- Custom answers to tell if from cache or not (see Sec. 3.2)
- Probe every 20min, for 2 to 3 hours
- Various TTLs: 60, 1800, 3600, and 86400s
- 15000 Vantage Points, 6800 R_n (no DDos)

Part 1: measuring caching in the wild



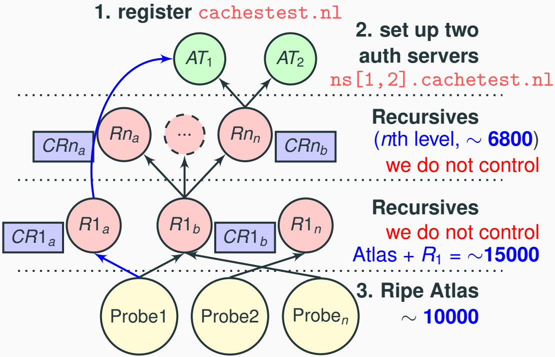
- Probes send unique queries to avoid cache interference
- Custom answers to tell if from cache or not (see Sec. 3.2)
- Probe every 20min, for 2 to 3 hours
- Various TTLs: 60, 1800, 3600, and 86400s
- 15000 Vantage Points, 6800 R_n (no DDos)

Part 1: measuring caching in the wild



- Probes send unique queries to avoid cache interference
- Custom answers to tell if from cache or not (see Sec. 3.2)
- Probe every 20min, for 2 to 3 hours
- Various TTLs: 60, 1800, 3600, and 86400s
- 15000 Vantage Points, 6800 R_n (no DDos)

Part 1: measuring caching in the wild

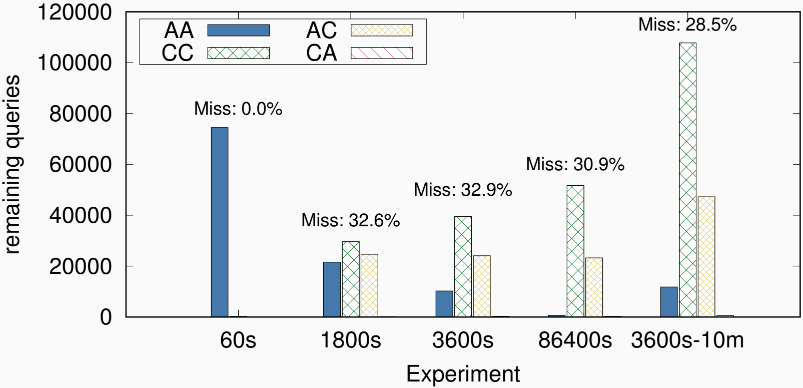


- Probes send unique queries to avoid cache interference
- Custom answers to tell if from cache or not (see Sec. 3.2)
- Probe every 20min, for 2 to 3 hours
- Various TTLs: 60, 1800, 3600, and 86400s
- 15000 Vantage Points, 6800 R_n (no DDos)

Part 1: measuring caching in the wild

- **How efficient is caching in the wild?**

Results: how good caching is in the wild?



- **Yellow color** is cache misses (AC)
- Good news: caching works fine for 70% of all 15,000 VPs
    - With our *not popular* domain
- but $\sim$ 30% of cache misses

Why cache misses (Why AC?)

Half of cache misses are from from complex caches like at Google

- cache fragmentation with multiple servers
- (previous work on Google DNS [6])

TTL	**60**	**1800**	**3600**	**86400**	**3600-10m**
AC Answers	37	24645	24091	23202	47,262
Public R_1	0	12000	11359	10869	21955
Google Public R_1	0	9693	9026	8585	17325
other Public R_1	0	2307	2333	2284	4630
Non-Public R_1	37	12645	12732	12333	25307
Google Public R_n	0	1196	1091	248	1708
other R_n	37	11449	11641	12085	23599

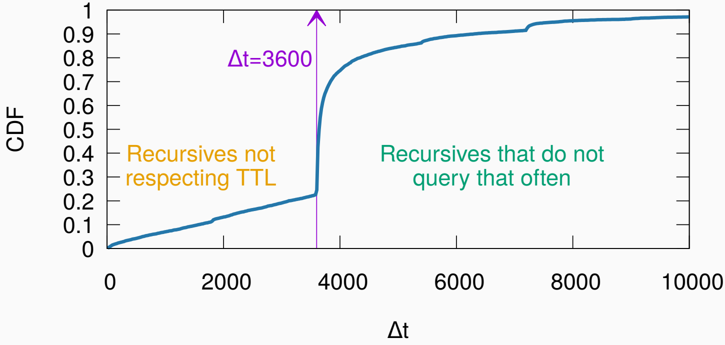
Table 1: AC answers (cache miss) public resolver classification

Part 2: caching in production zones

- Caching works 70% as expected
- **Are these experiments representative?**
- We look at `.nl` production data
    - we compute Δt (time since last query)
    - Compare to TTL of 3600s
    - 485k queries from 7,779 recursives

Part 2: caching in production zones

- Most resolvers send queries usually ~3600s (`.nl` TTL)
- 28% do not respect the 1h TTL
- **Yes, experiments are like real zone**
- (we also look into the Roots , see paper [4])

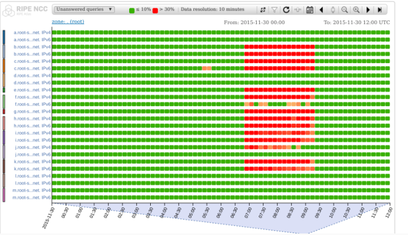


OK, so what do you we have so far?

- We know how caching works in the wild (both Ripe and `.nl`)
- Time to move Part 3: **What happens under DDoS attacks?**
- Goal: understand client experience under DDoS

Part 3: Emulating DDoS

Root DNS DDoS Nov 2015



RIPE NCC

Unanswered queries ▼ 🟩 ≤ 10% 🟥 > 30% | Data resolution: 10 minutes

zone: . (root)

From: 2015-11-30 00:00 To: 2015-11-30 12:00 UTC

Dyn Oct 2016

Hackers Used New Weapons to Disrupt Major Websites Across U.S.

- Remember: clients experience varied significantly for these
- Our goal is to explain their experience

Part 3: Emulating DDoS

- Similar setup as other experiments:
- Emulate DDoS: drop incoming queries at certain rates at Authoritative servers, with `iptables`

Complete DDoS

- 100% packet loss via `iptables`
- TTL=3600s (1 hour)
- We probe every 10 minutes
- At $t = 10min$, we drop all packets

Complete DDoS: TTL: 60min, 100% failure

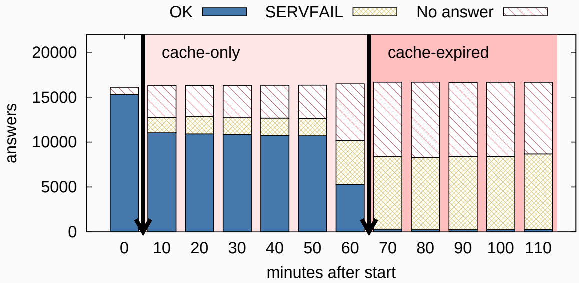


Figure 1: Experiment A: 100% failure after 10min, TTL: 60min

- DDoS starts after 1st query (fresh cache)
- During DDoS: **70% of clients are served** ☺ (cache)
    - except right at 60min (expire)
- After cache expires: only 0.2% clients (serve state)
    - `draft-ietf-dnsop-serve-stale-02`

Complete DDoS: changing cache freshness

- Prior experiment had **OPTIMAL** cache, loaded just before attack
- Now we load the cache much **earlier**

Complete DDoS: changing cache freshness

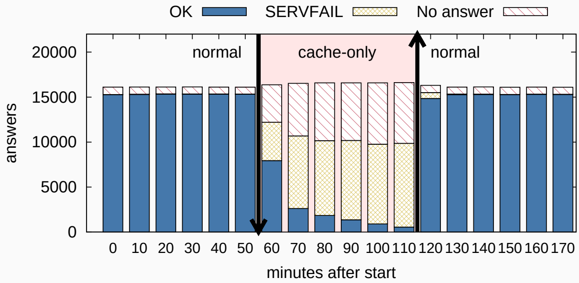


Figure 2: Experiment B: 100% failure after 60min, TTL: 60min

- Cache much less effective (most users ☹)
- Why? TTL is **decremented** over time in caches

Complete DDoS: changing cache freshness

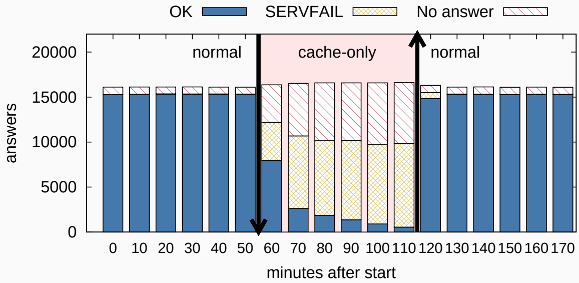


Figure 2: Experiment B: 100% failure after 60min, TTL: 60min

- Cache much less effective (most users ☹)
- Why? TTL is **decremented** over time in caches

Complete DDoS: changing TTL

- Caching freshness impacts user experience
- **How TTL impacts clients' experience?**

Complete DDoS: TTL influence

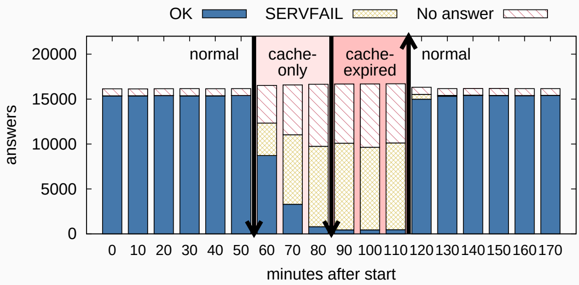


Figure 3: Experiment C: 100% failure after 60min, TTL: 30min

- Users experience worsens with shorter TTL
- Most users ☹

Complete DDoS: User Experience Discussion

- caching helps 70% of cases
- caches don’t work after they time out
  - except for serve slate
- caches will time-out at different times
- conclusion:
  - operators with modest TTLs get quite a bit of protection
  - serve-stale would help

Partial Failure DDoS

- Not all DDoS are strong enough to bring all servers down
- Some lead to partial failure (Root DNS Nov 2015 [3])

- **In this case, how would users experience the attack?**

Partial Failure DDoS: 50% success

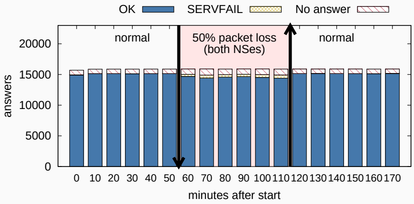
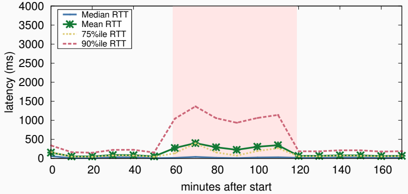


Figure 4: Experiment E: 50% failure after 60min, TTL: 60min



Good: most clients get answer ☺ , even at 50% loss

- but more latency

Partial Failure DDoS: changing intensity

- Let’s emulate an attack that leads to 90% packet loss
- **How will that impact clients experience?**

Partial Failure DDoS: changing intensity to 90%

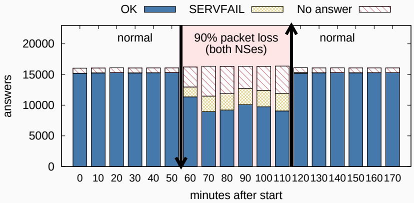
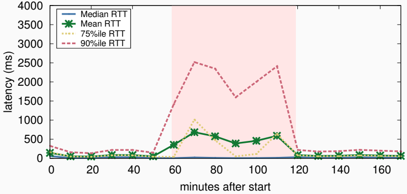


Figure 5: Experiment H: 90% success DDoS, TTL: 30min



Good: most clients STILL get answer ☺, even at **90%** loss (but more latency)

Partial Failure DDoS: disabling caching

- TTL = 1 minute
- Probing Interval = 10minutes
    - Cache expires before new round of measurements
- Emulates CDNs setup
- We drop 90% of packets

Partial Failure DDoS: disabling caching

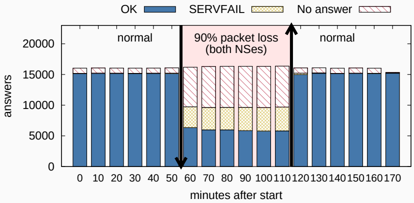
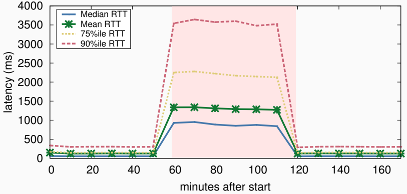


Figure 6: Experiment I: 90% success DDoS, TTL: 1min



- Even with no caching (TTL 1min), 27% get an answer ☺
- Most users ☹

Partial Failure DDoS: recursives retrying

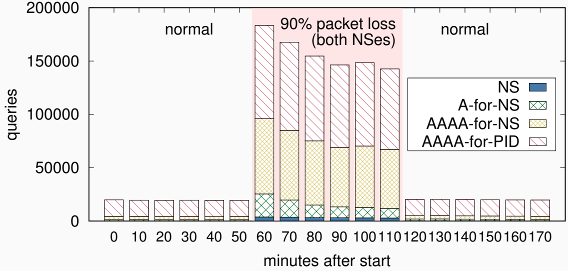


Figure 7: Queries received at Auth Servers for Experiment I: 90% success DDoS, TTL: 1min

- Part of DNS resilience is that recursives keep on retrying
- Recursives will “hammer” authoritatitve servers
- **Friendly fire 8.1x** in case of no caching

Partial Failure DDoS: more recursives in use

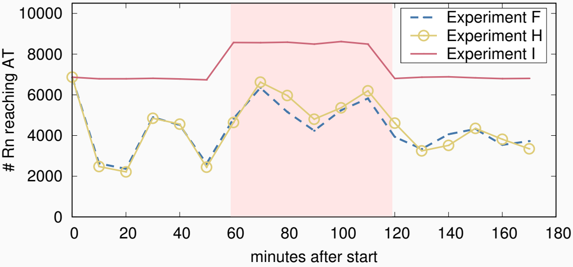


Figure 8: Unique *Rn* recursives addresses observed at authoritatives

- We have ∼15k vantage points and ∼6.8k R_n recursives
- Partial DDoS: R_n increases to 8.5k (24%) on Exp. I
- Shows complex recursives infrastructure; more are used in case of failure

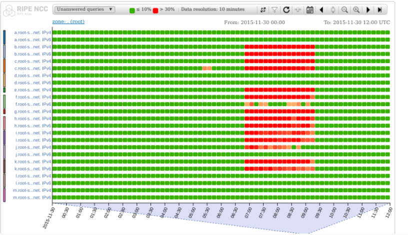
Partial Failure DDoS: User Experience Discussion

- Recursive infrastructure will “expand” and retry
    - More recursives in use seen at authoritatives
    - Same recursives will retry multiple times
- **Users** may experience longer latency
    - As recursives will retry to resolve the domain
- Caching reduces latency during DDoS
- The longer the TTL, the better the user experience
    - provided caches are filled and not about to expire

Implications

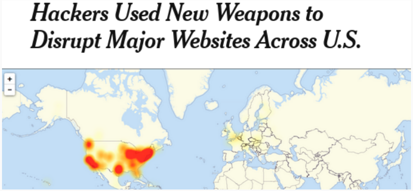
Our experiments explain user's experiences in previous DDoS

Root DNS DDoS Nov 2015



- Users: no known reports of errors
- Why? **Longer TTLs** and some servers remained up

Dyn Oct 2016



Hackers Used New Weapons to Disrupt Major Websites Across U.S.

- Users: many could not resolve
- Why? **Shorter TTLs** and others

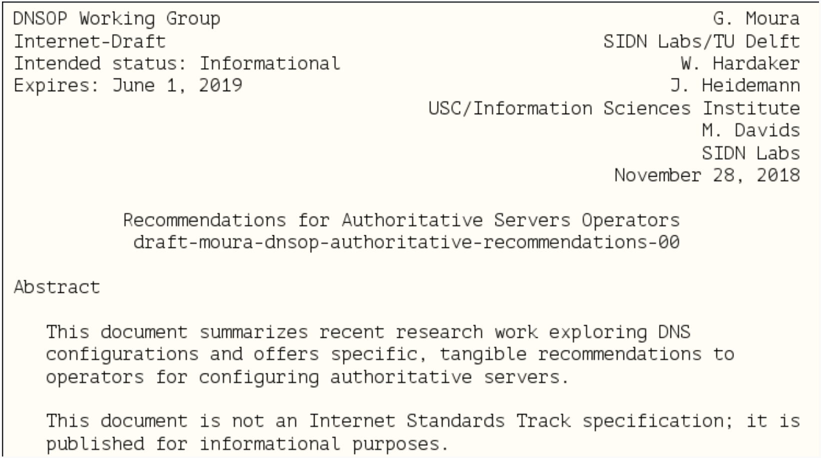
Conclusions

- **Caching and retries**: important part of DNS resilience
  - 50-60% clients served with 90% packet loss (TTL 30min)
  - 27% clients served with 90% packet loss (TTL 1min)
- Explain recent DDoS outcomes
- **What’s the “best TTL" ?**
  - There’s a clear trade-off between TTL and DDoS robustness, choose longer if you can
  - There’s no “one size fits all” solution
- **IETF draft (hopefully to be adopted by DNSOP)**

  `draft-moura-dnsop-authoritative-recommendations-00`

contact: `giovane.moura@sidn.nl`

IETF draft



```
DNSOP Working Group                                         G. Moura
Internet-Draft                                   SIDN Labs/TU Delft
Intended status: Informational                           W. Hardaker
Expires: June 1, 2019                                    J. Heidemann
                                   USC/Information Sciences Institute
                                                            M. Davids
                                                            SIDN Labs
                                                    November 28, 2018


          Recommendations for Authoritative Servers Operators
           draft-moura-dnsop-authoritative-recommendations-00

Abstract

   This document summarizes recent research work exploring DNS
   configurations and offers specific, tangible recommendations to
   operators for configuring authoritative servers.

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.
```

References I

[1] Sam Kottler.

February 28th DDoS Incident Report | Github Engineering, March 2018.

. https://githubengineering.com/ddos-incident-report/.

[2] Carlos Morales.

February 28th DDoS Incident Report | Github EngineeringNETSCOUT Arbor Confirms 1.7 Tbps DDoS Attack; The Terabit Attack Era Is Upon Us, March 2018.

https://www.arbornetworks.com/blog/asert/netscout-arbor-confirms-1-7-tbps-ddos-attack-terabit-att

References II

[3] Giovane C. M. Moura, Ricardo de O. Schmidt, John Heidemann, Wouter B. de Vries, Moritz Müller, Lan Wei, and Christian Hesselman.

Anycast vs. DDoS: Evaluating the November 2015 root DNS event.

In *Proceedings of the ACM Internet Measurement Conference*, November 2016.

References III

[4] Giovane C. M. Moura, John Heidemann, Moritz Müller, Ricardo de O. Schmidt, and Marco Davids.

When the dike breaks: Dissecting DNS defenses during DDoS (extended).

In *Proceedings of the ACM Internet Measurement Conference*, October 2018.

[5] Nicole Perlroth.

Hackers used new weapons to disrupt major websites across U.S.

New York Times, page A1, Oct. 22 2016.

References IV

[6] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman.

On measuring the client-side DNS infrastructure.

In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 77–90. ACM, October 2013.