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]

OARC 29, Amsterdam, The Netherlands
2018-10-14

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

Research paper to appear on ACM IMC 2018

- Joint research work to appear at:
  https://conferences.sigcomm.org/imc/2018/
- Full text (PDF):
  https://www.isi.edu/~johnh/PAPERS/Moura18b.pdf

DDoS Attacks

- DDoS attacks are on the rise
- Getting bigger, more frequent, cheaper, and easier
  - Arbor: 1.7 Tb/s [2] (2018)
  - Github DDoS: 1.35 Tb/s [1] (2018)
  - Dyn DDoS: 1.2 Tb/s (Mirai IoT) [6] (2017)
  - DDoS as a service: few dollars with booters [8].
- Many DNS services have been victim of DDOS attacks

DDoS and DNS: two examples

Root DNS DDoS Nov 2015

no known reports of errors seen by users [3]

Dyn Oct 2016

some users could not reach popular sites [6]

Two large DDoSes, very different outcomes. Why?

DDoS and DNS: two examples

Root DNS DDoS Nov 2015

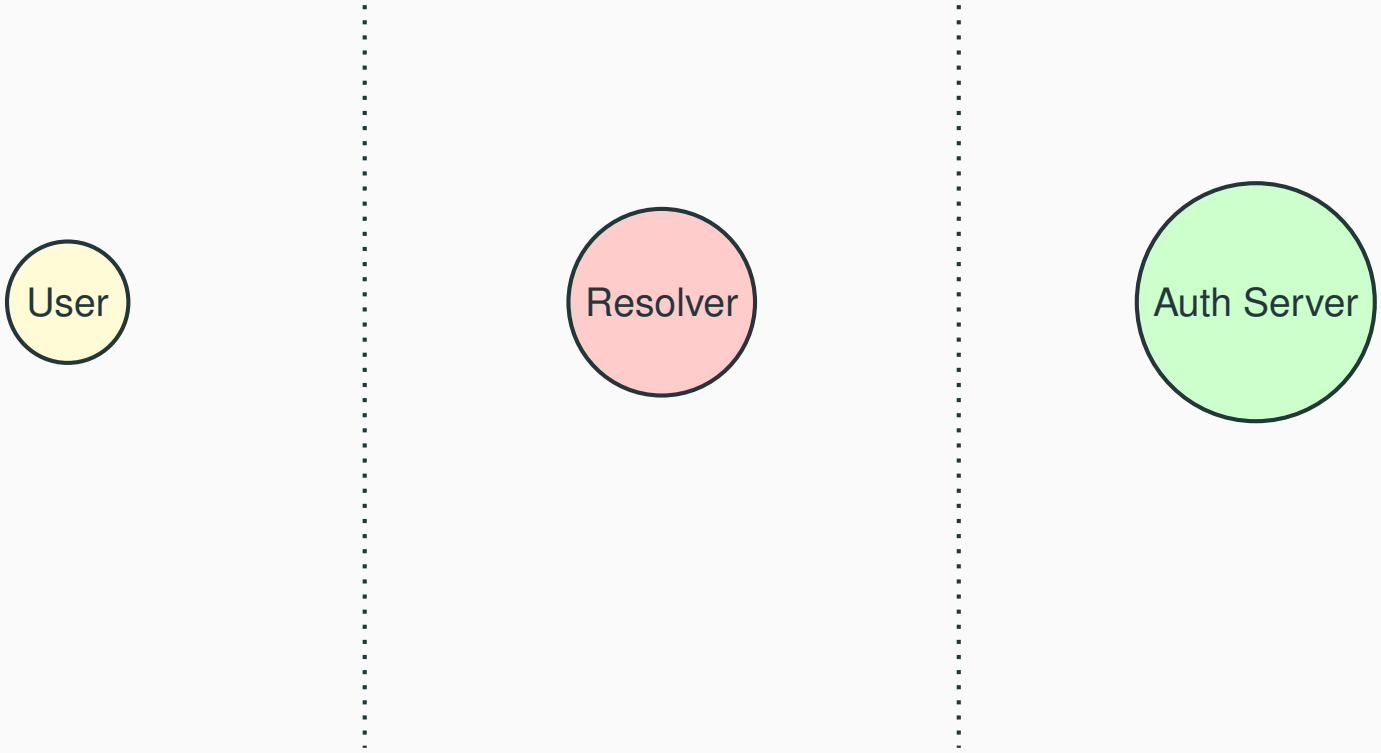
no known reports of errors seen by users [3]

Dyn Oct 2016

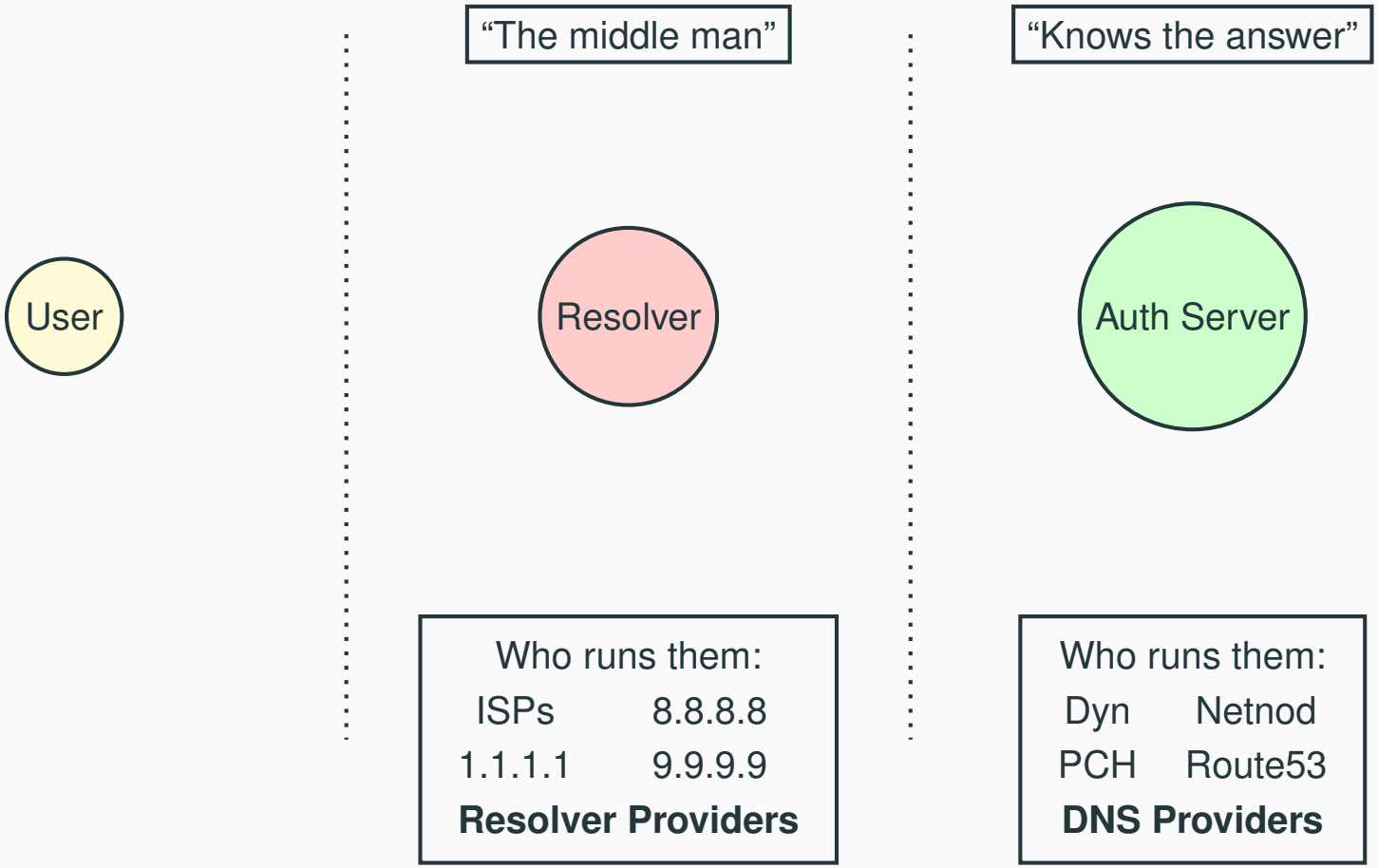
some users could not reach popular sites [6]

Two large DDoSes, very different outcomes. Why?

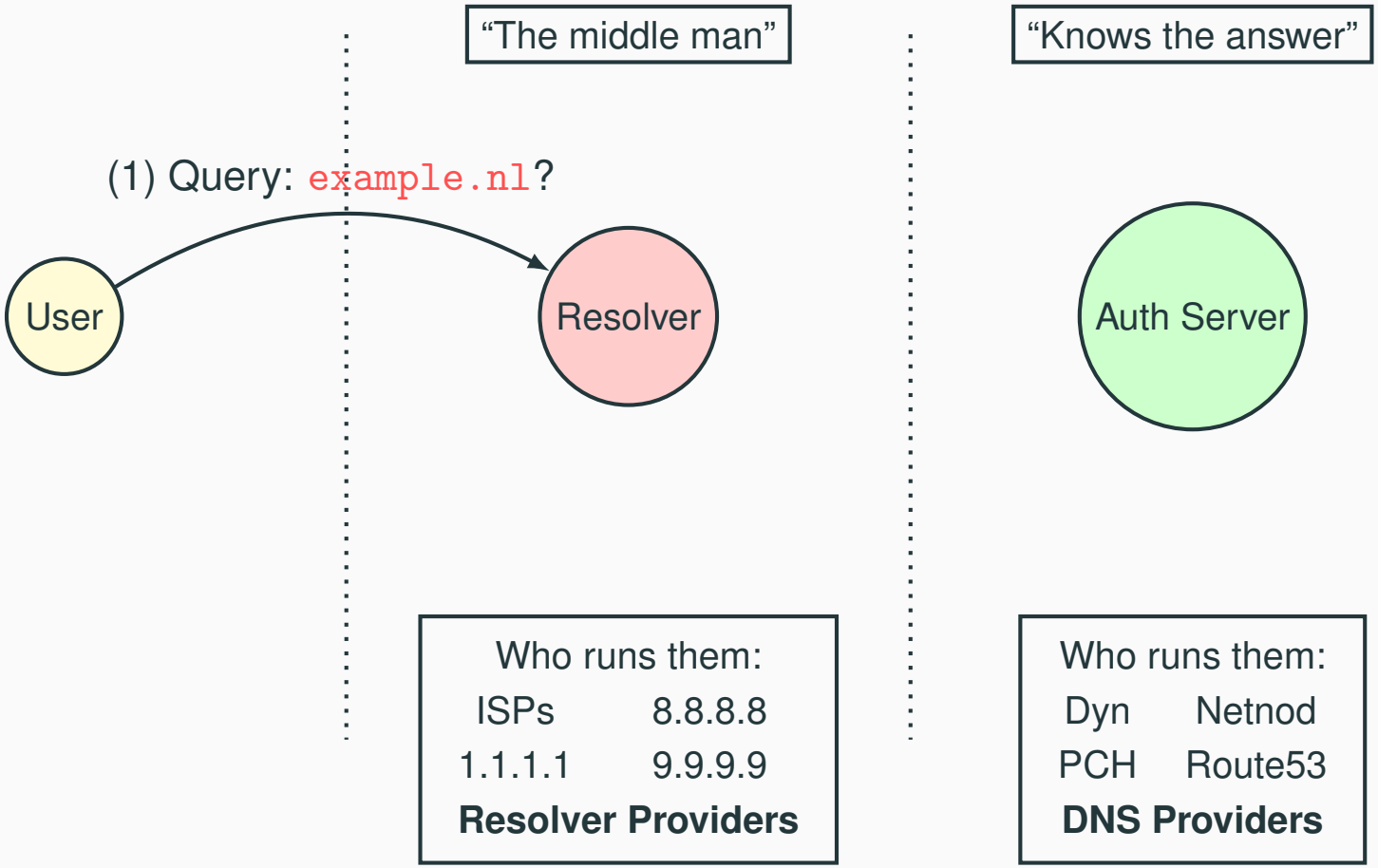
DNS Basics



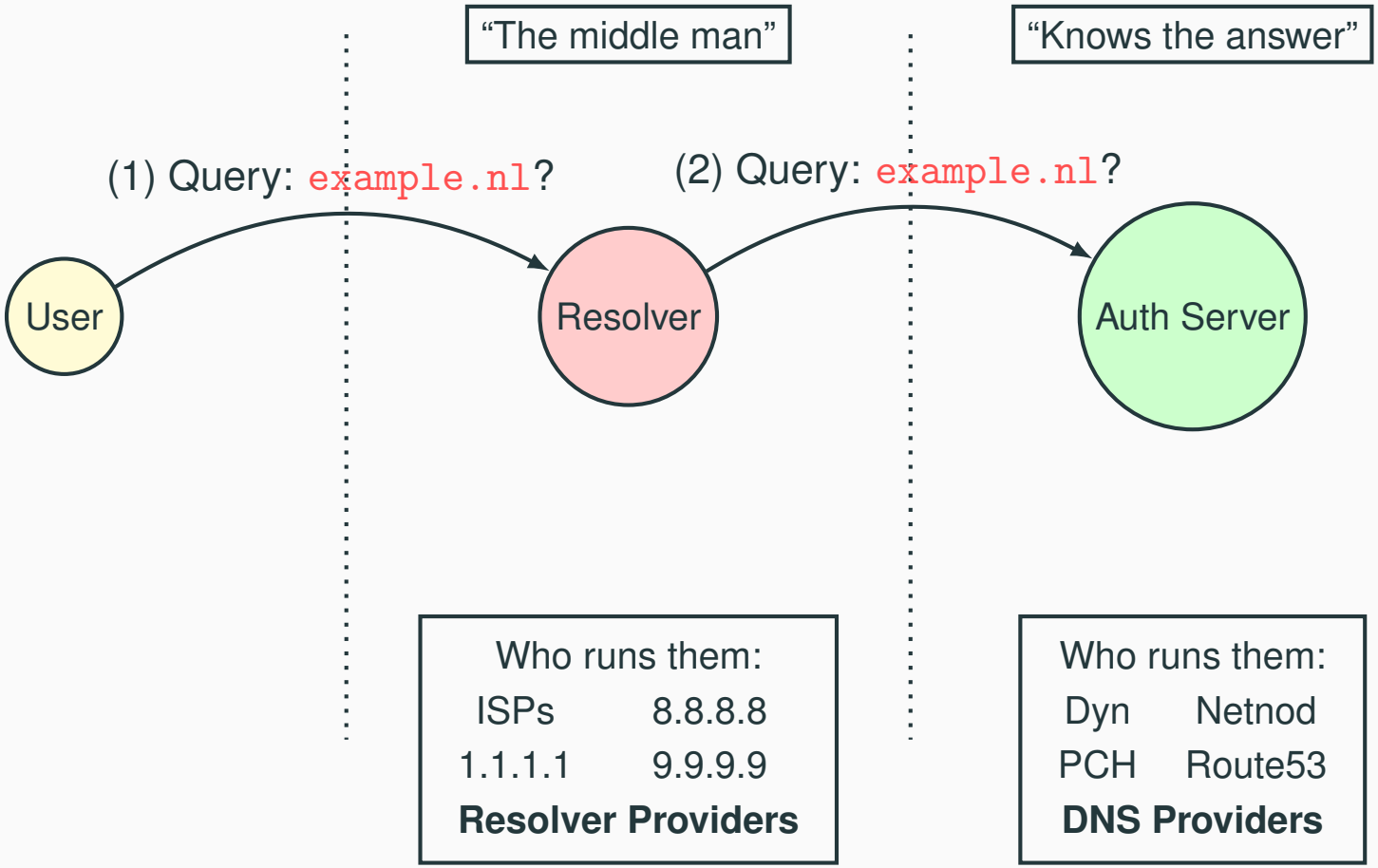
DNS Basics


“The middle man”
“Knows the answer”
User
Resolver
Auth Server
Who runs them:
ISPs 8.8.8.8
1.1.1.1 9.9.9.9
Resolver Providers
Who runs them:
Dyn Netnod
PCH Route53
DNS Providers

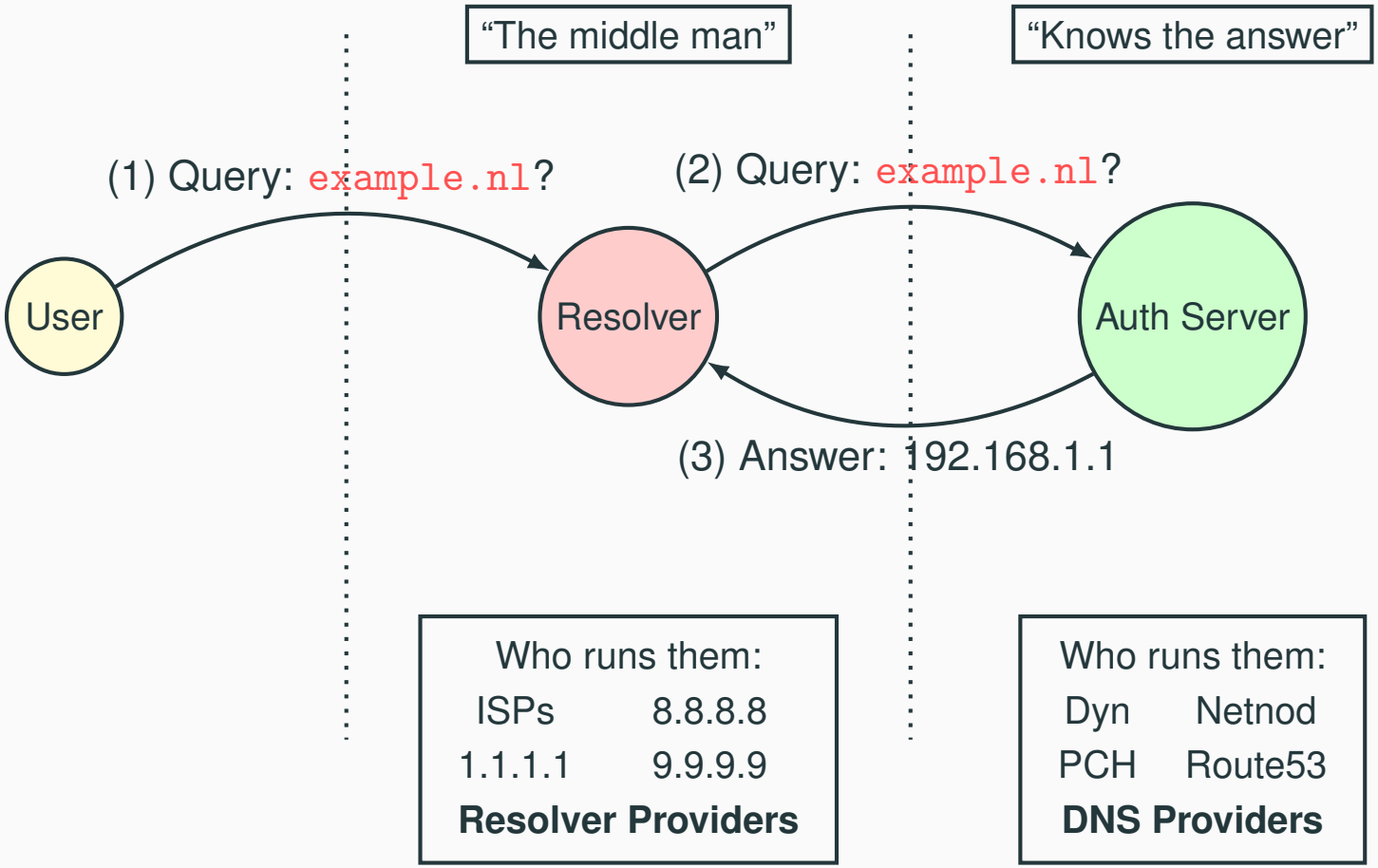
DNS Basics


“The middle man”
“Knows the answer”
(1) Query: example.nl?
User
Resolver
Auth Server
Who runs them:
ISPs 8.8.8.8
1.1.1.1 9.9.9.9
Resolver Providers
Who runs them:
Dyn Netnod
PCH Route53
DNS Providers

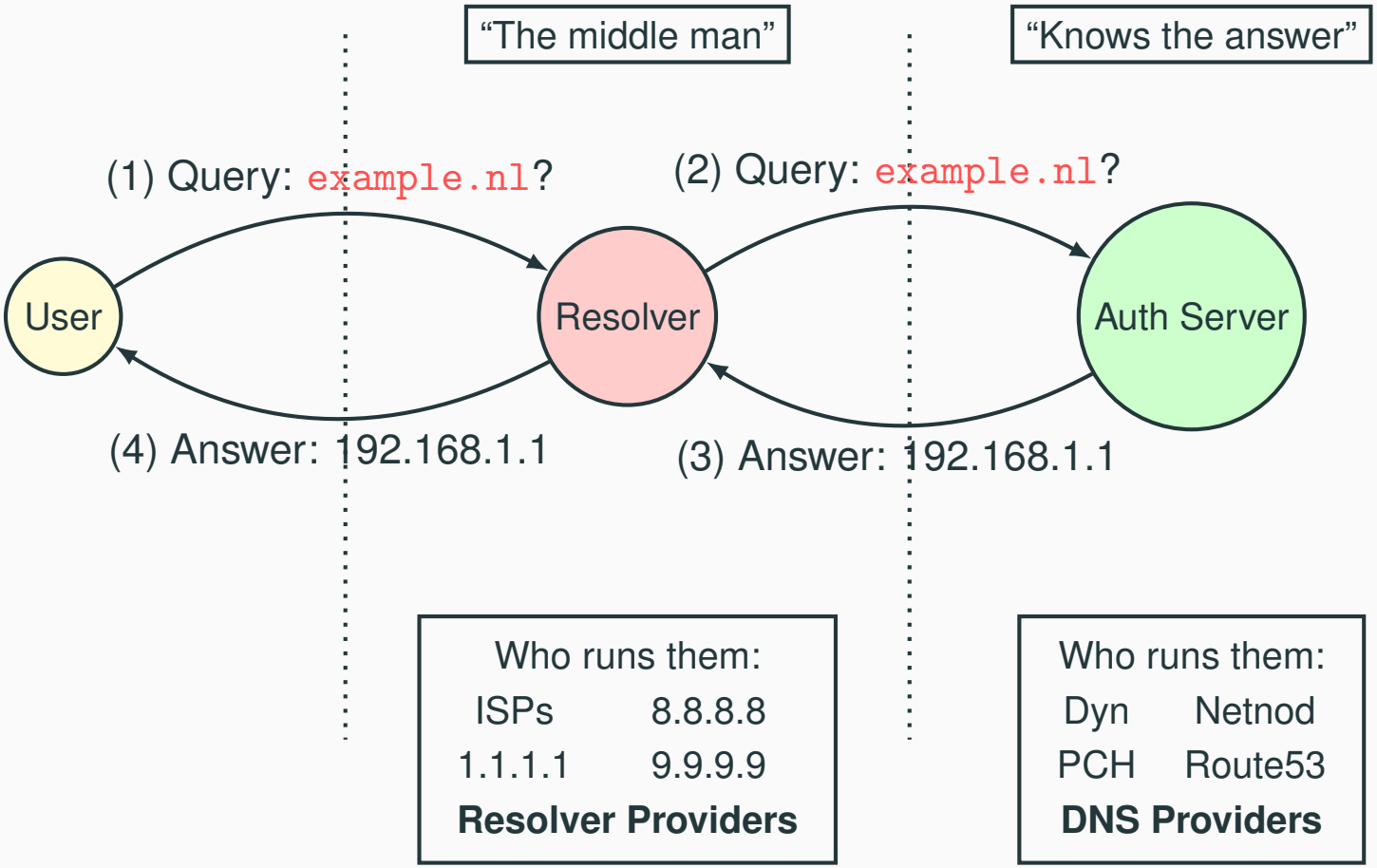
DNS Basics


“The middle man”
“Knows the answer”
(1) Query: example.nl?
(2) Query: example.nl?
User
Resolver
Auth Server
Who runs them:
ISPs 8.8.8.8
1.1.1.1 9.9.9.9
Resolver Providers
Who runs them:
Dyn Netnod
PCH Route53
DNS Providers

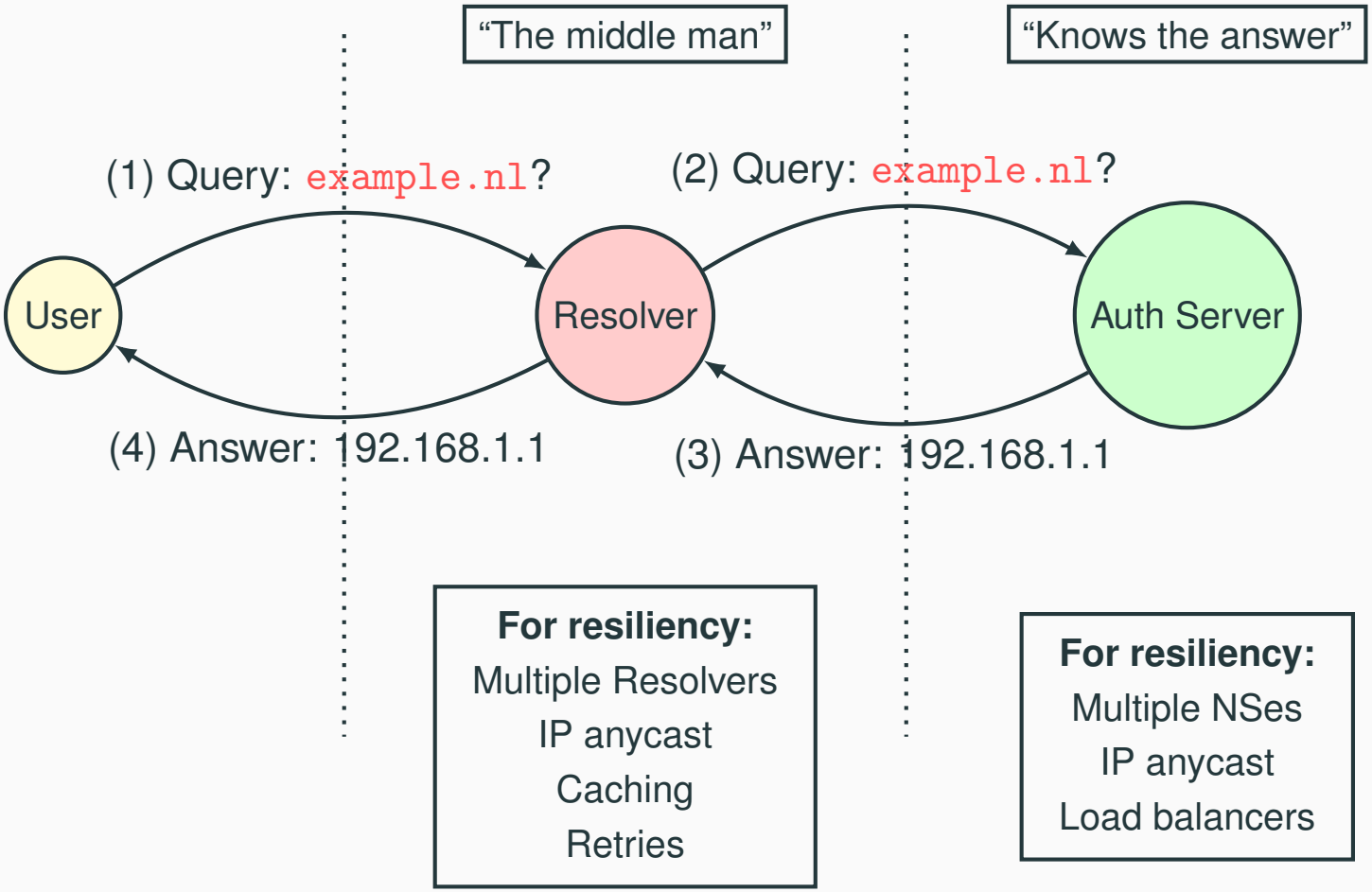
DNS Basics


“The middle man”
“Knows the answer”
(1) Query: example.nl?
(2) Query: example.nl?
User
Resolver
Auth Server
(3) Answer: 192.168.1.1
Who runs them:
ISPs 8.8.8.8
1.1.1.1 9.9.9.9
Resolver Providers
Who runs them:
Dyn Netnod
PCH Route53
DNS Providers

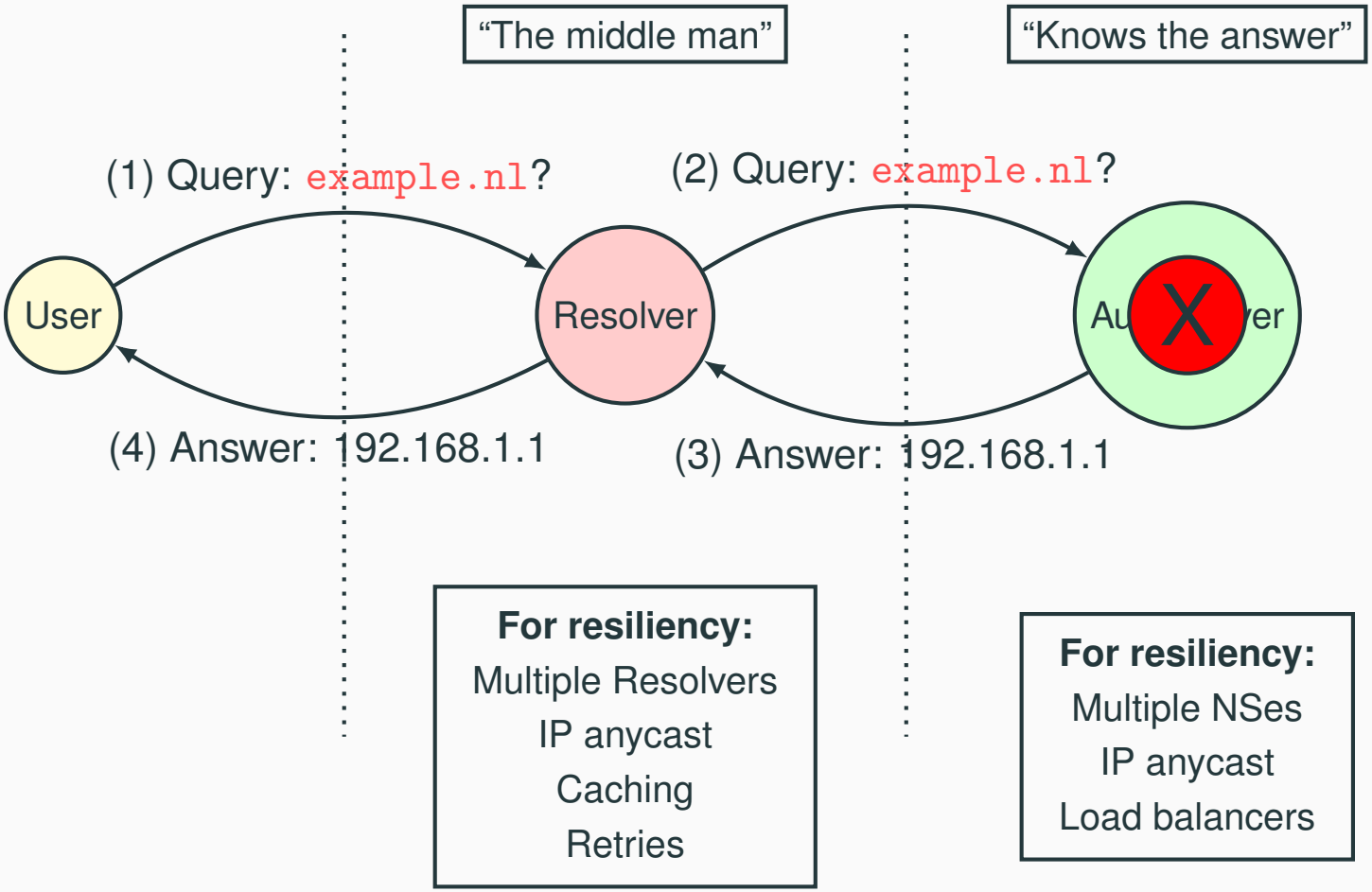
DNS Basics


“The middle man”
“Knows the answer”
(1) Query: example.nl?
(2) Query: example.nl?
User
Resolver
Auth Server
(4) Answer: 192.168.1.1
(3) Answer: 192.168.1.1
Who runs them:
ISPs 8.8.8.8
1.1.1.1 9.9.9.9
Resolver Providers
Who runs them:
Dyn Netnod
PCH Route53
DNS Providers

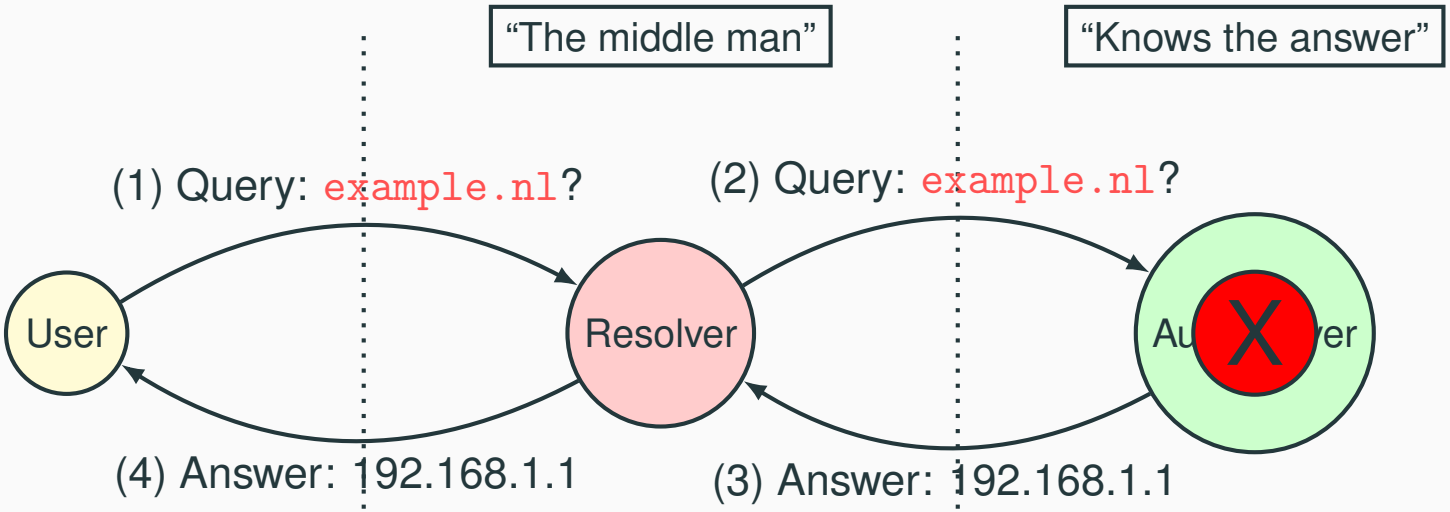
DNS Basics


“The middle man”
“Knows the answer”
(1) Query: example.nl?
(2) Query: example.nl?
User
Resolver
Auth Server
(4) Answer: 192.168.1.1
(3) Answer: 192.168.1.1
For resiliency:
Multiple Resolvers
IP anycast
Caching
Retries
For resiliency:
Multiple NSes
IP anycast
Load balancers

DNS Basics


“The middle man”
“Knows the answer”
(1) Query: example.nl?
(2) Query: example.nl?
User
Resolver
X
(4) Answer: 192.168.1.1
(3) Answer: 192.168.1.1
For resiliency:
Multiple Resolvers
IP anycast
Caching
Retries
For resiliency:
Multiple NSes
IP anycast
Load balancers

DNS Basics



How will the clients be affected?

For resiliency:
Multiple Resolvers
IP anycast
Caching
Retries

For resiliency:
Multiple NSes
IP anycast
Load balancers

How much will resolvers help?

Evaluating DNS Resiliency

- **Part 1**: evaluate user experience under “normal” operations
- **Part 2**: Verify results of Part 1 in production zones (`.nl`)
- **Part 3**: Emulate DDoSes in the wild to evaluate caching/retrials under stress, **to observe user experience**

Part 1: measuring caching in the wild

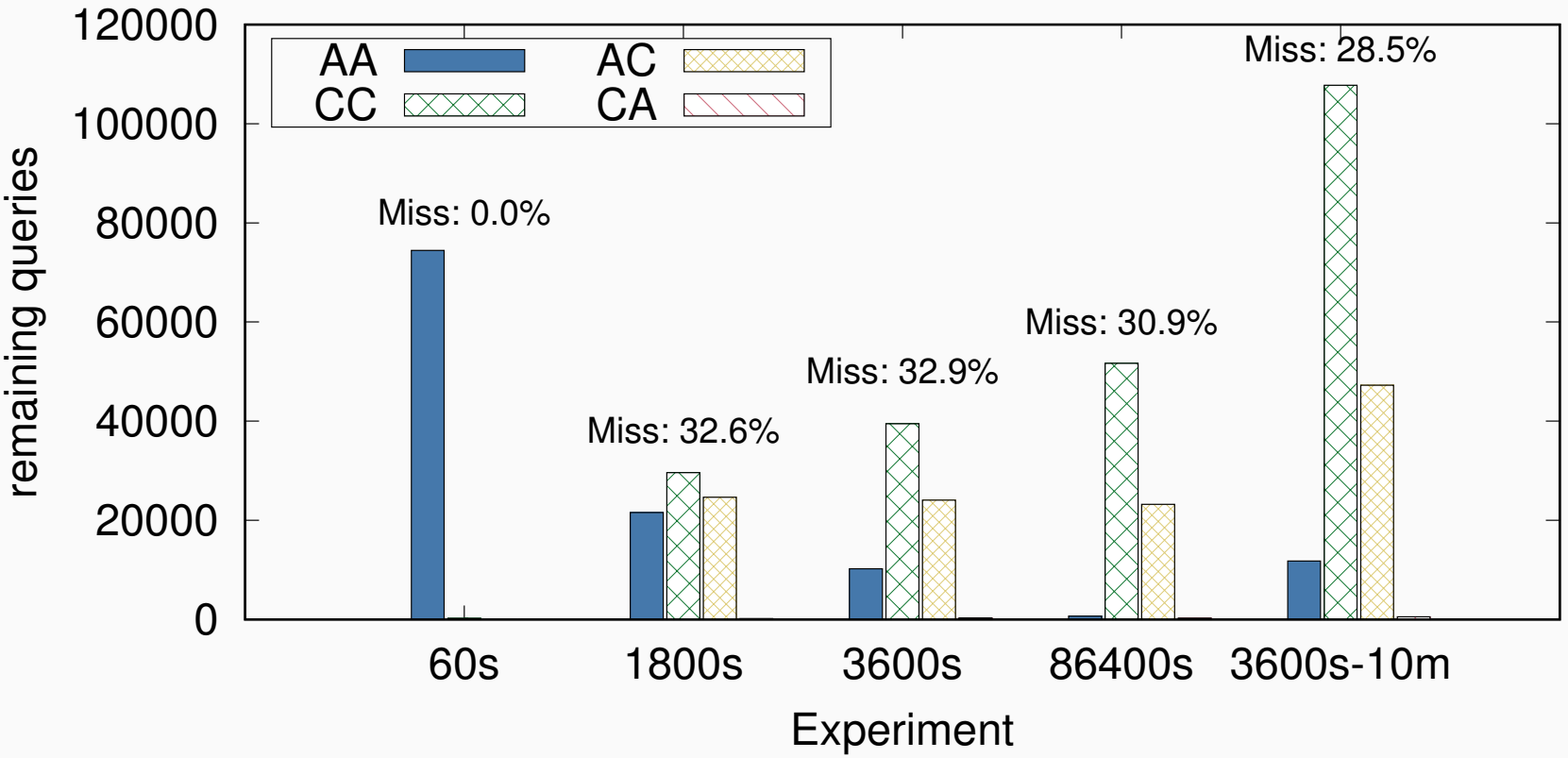
Setup

1. register our new domain (`cachetest.nl`)
2. run two unicast IPv4 authoritatives on EC2 Frankfurt
3. User Ripe Atlas and their resolvers as vantage points ($\sim$ 15k)
4. Each VP sends a unique AAAA query, so no interference
   - e.g.,: `500.cachetest.nl` for probeID=500
5. Each AAAA DNS answer encodes a counter that allow us to tell if it was cache hit or miss
   - `$PREFIX:$SERIAL:$PROBEID:$TTL`
6. Probe every 20min, and run scenarios with different TTLs, for 2 to 3 hours (to match various TTLs in the wild)
   - 60, 1800,3600, and 86400 seconds TTL

Part 1: measuring caching in the wild

- We control auth servers and clients (stub resolver)
- We do not control recursives
- How efficient is caching in the wild?
    - Remember: TTL sets upper limit for HOW LONG it should be cached by recursives

Results: how good caching is in the wild?



1. Good news: caching works fine for 70% of all 15,000 VPs
   - With our *not popular* domain
2. Not so good news: $\sim$ 30% of cache misses (AC)

Why cache misses (Why AC?)

Possible: capacity limits, cache flushes, complex caches

Mostly: complex caches

- cache fragmentation with multiple servers
- (previous work on Google DNS [9])

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

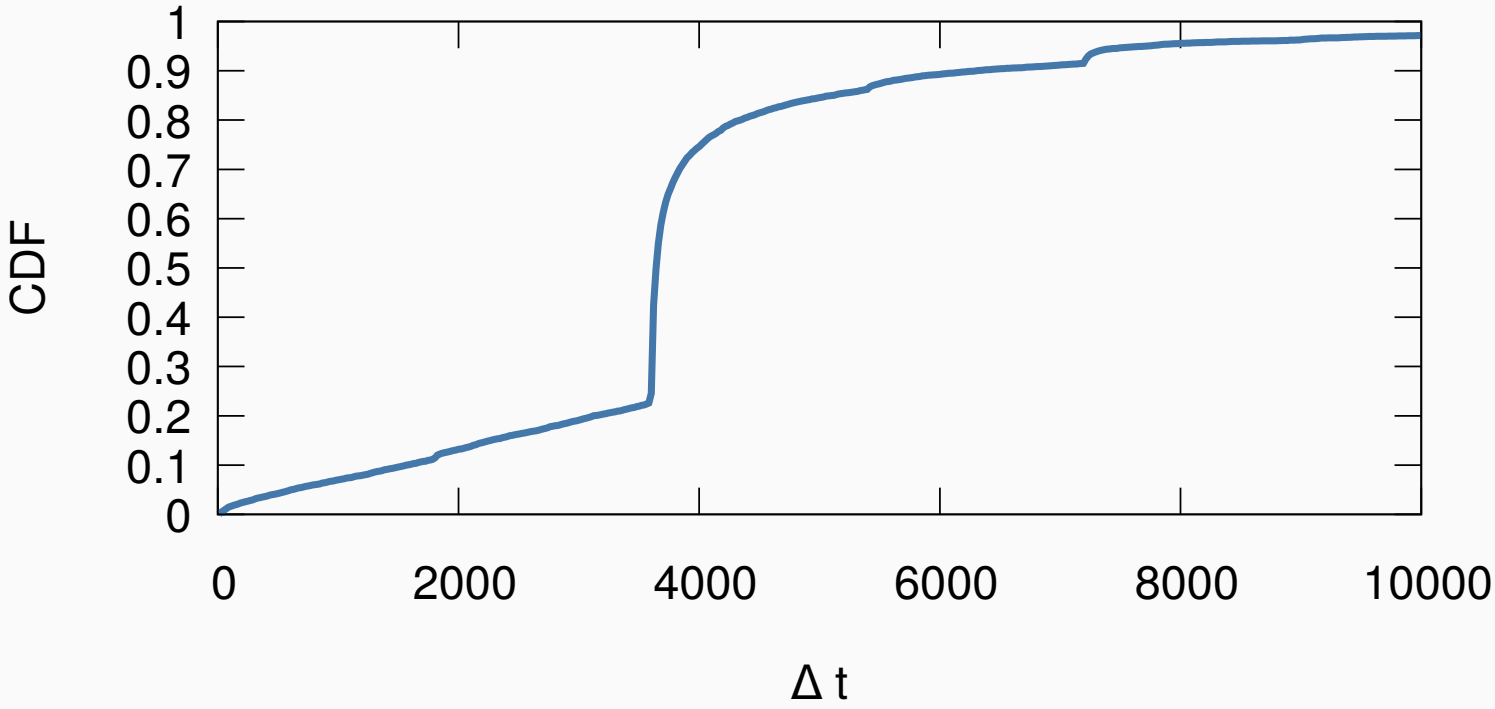
Table 1: AC answers (cache miss) public resolver classification

Part 2: caching in production zones

- OK, in our controlled environment, we show that 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: emulate DDoS
- Goal: understand client experience under DDoS

Part 3: Emulating DDoS

- Similar setup as other experiments
- Emulate DDoS: drop incoming queries at certain rates at Authoritative servers, with `iptables`
- Question: (when) do caches protect clients?
- Or why some DDoS attacks seem to have more impact?
- We show only few experiments, many more in the paper

Scenario A: all servers DOWN

- Worst **nightmare** for a DNS operator
- Only resolver's cache can save clients
- TTL=3600s (1 hour)
- We probe every 10 minutes
- At $t = 10min$, we drop all packets

Complete DDoS: TTL: 60min, 100% failure

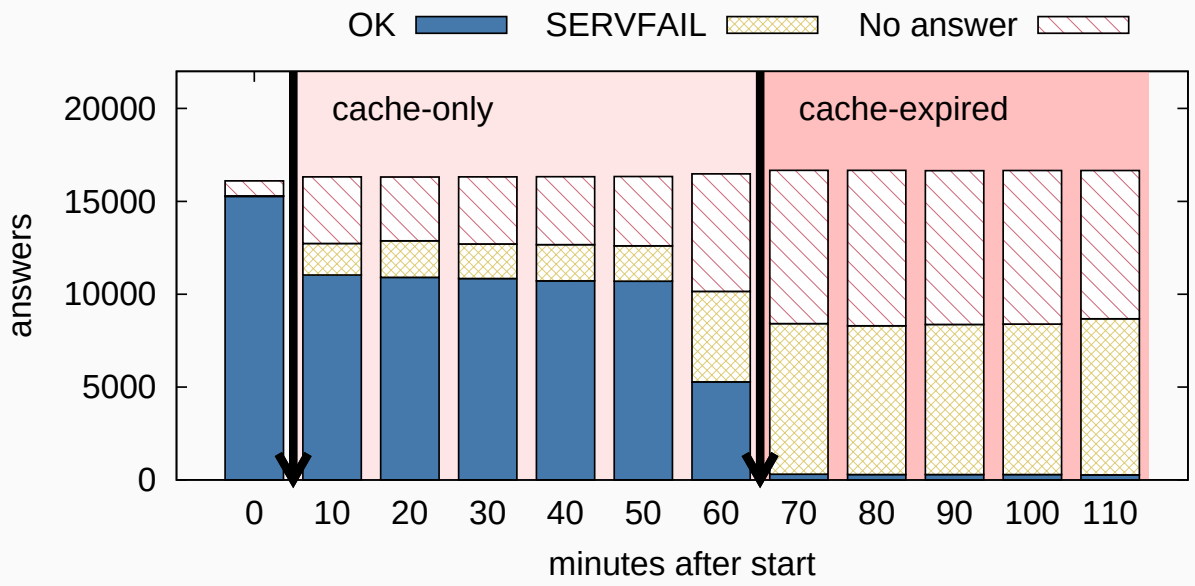


Figure 1: Scenario A: 100% failure after 10min, TTL: 60min

- DDoS starts after 1st query (fresh cache)
- During DDoS: **35%-70% of clients are served** (cache)
- After cache expires: only 0.2% clients (serve state)
  - `draft-ietf-dnsop-serve-stale-00`

Complete DDoS: changing cache freshness

- Scenario B: Cache freshness: about to expire
- How clients will experience DDoS?

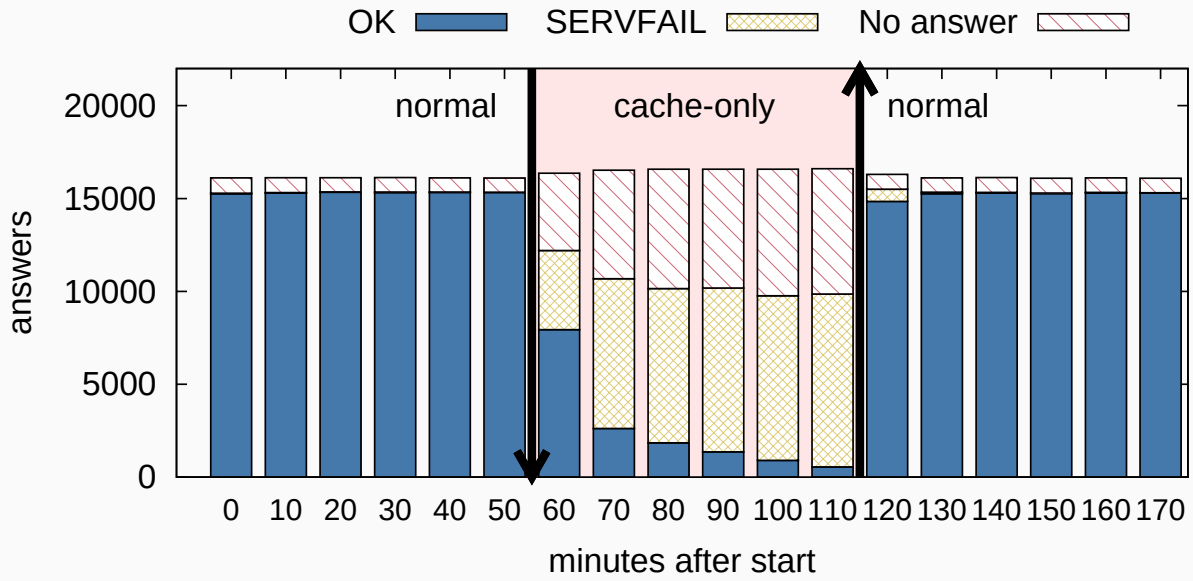


Figure 2: Scenario B: 100% failure after 60min, TTL: 60min

- Cache much less effective (as times out near attack)
- Fragmented cached helps some (by filling later)

Complete DDoS: changing cache freshness

- Scenario B: Cache freshness: about to expire
- How clients will experience DDoS?

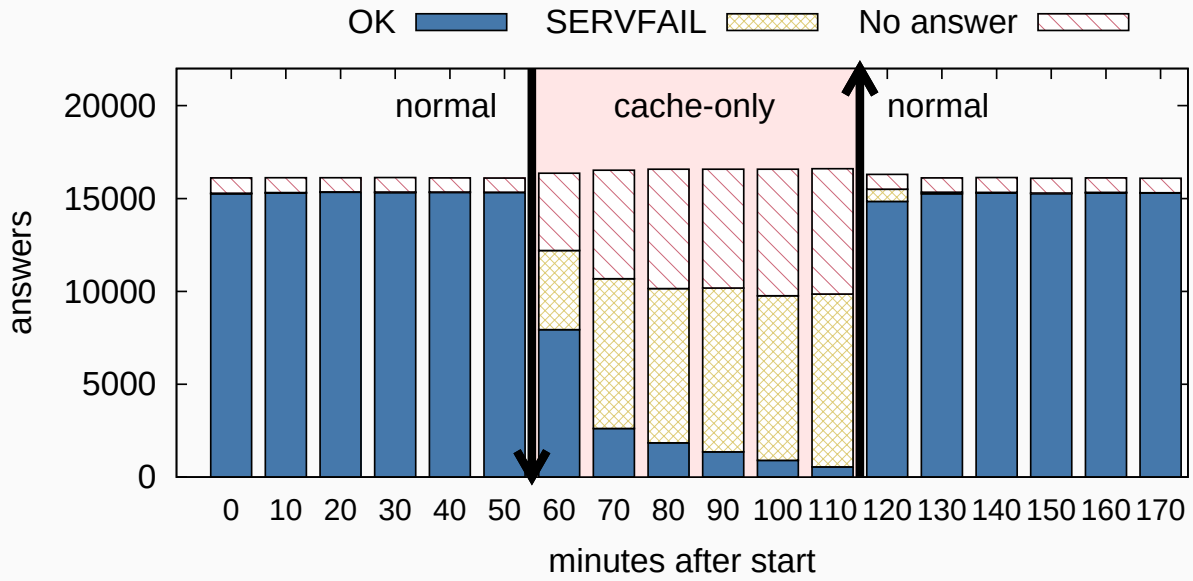


Figure 2: Scenario B: 100% failure after 60min, TTL: 60min

- Cache much less effective (as times out near attack)
- Fragmented cached helps some (by filling later)

Complete DDoS: TTL record influence

- Influence of TTL: reducing from 60min to 30min
- How clients will experience DDoS?

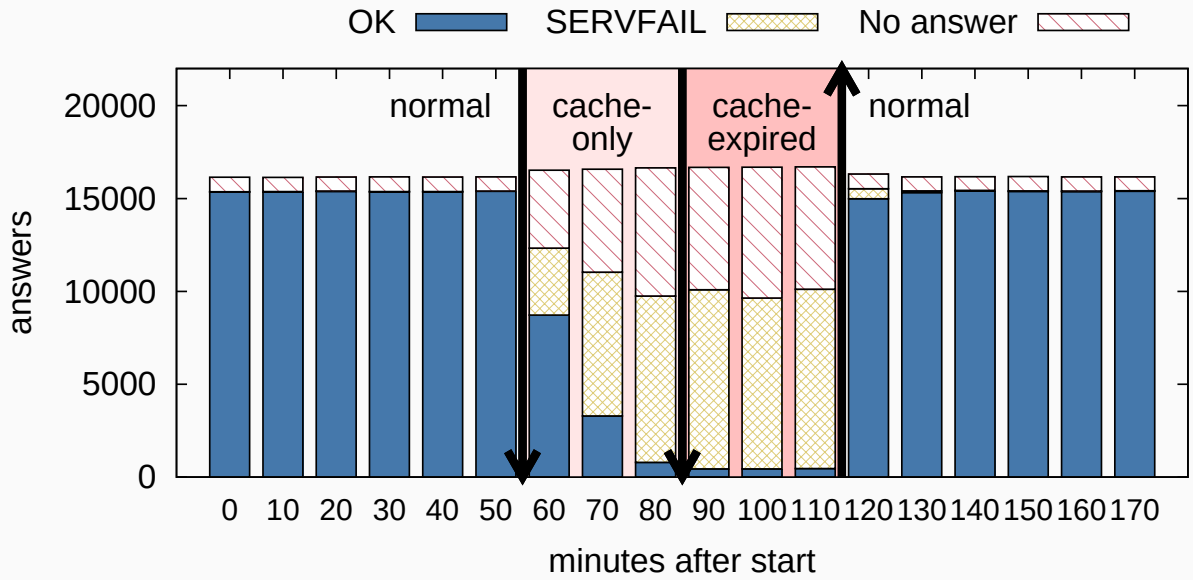


Figure 3: Scenario C: 100% failure after 60min, TTL: 30min

- Users experience worsens with shorter TTL
- OPs: choose wisely the TTL of your records when engineering for DDoS

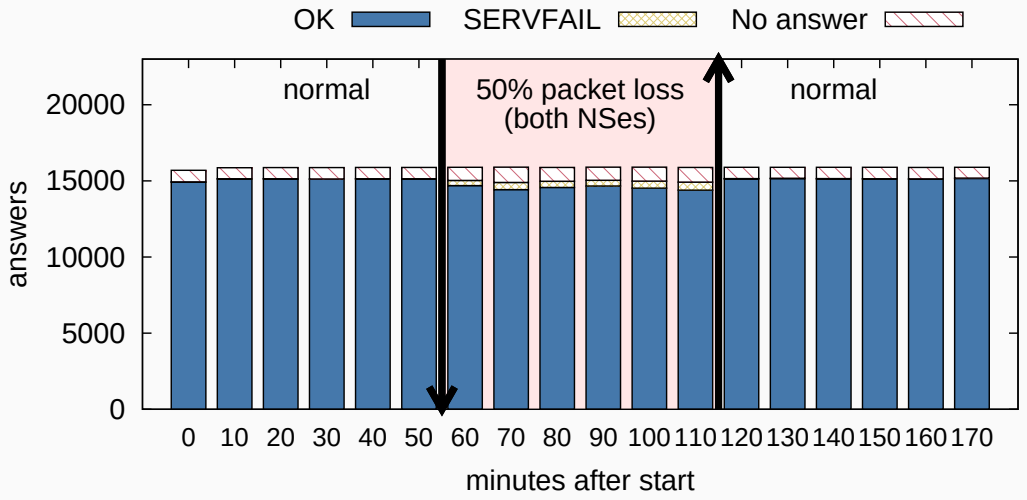
Discussion complete DDoS

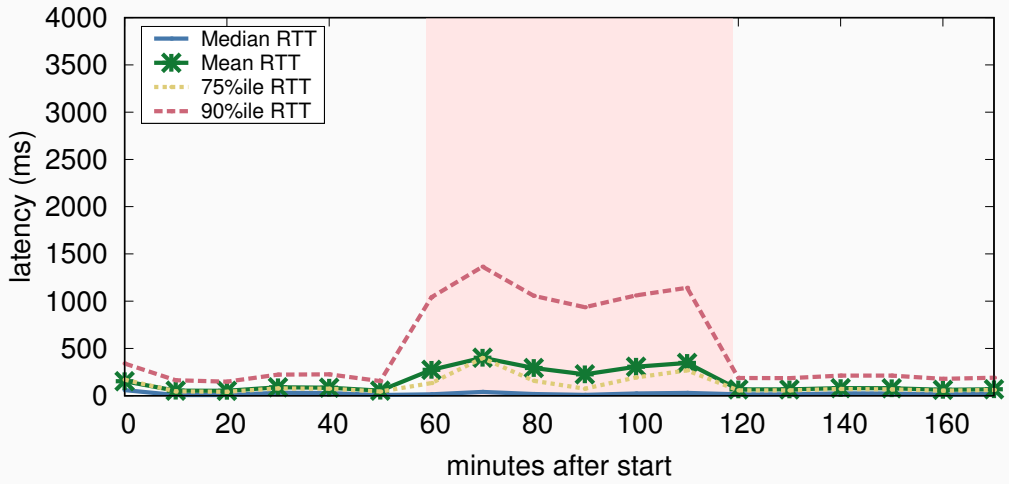
- Caching is *partially* successful during complete DDoS
- OPs: don't expect protection for clients as long as your TTL; depends on their cache state
- Serving stale content provides the last resort for Doomsday scenario
  - some ops (Google, OpenDNS) seem to do it, but it is not widespread yet
- TTL of records: the shorter you set them, the less you protect users during a complete DDoS

Partial DDoS

- Not all DDoS are strong enough to bring all servers down
- Some lead to partial failure (Root DNS Nov 2015 [3])
  - Partial failure: some of the available authoritative fail to answer all queries, or take longer to answer; then users experience longer latencies
- In this case, how would users experience the attack?

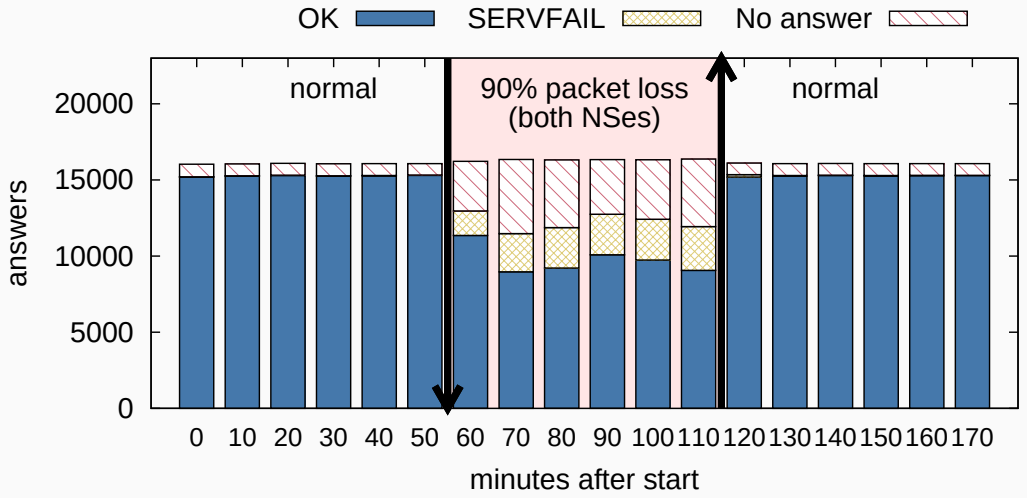
Experiment E: 50% success DDoS, TTL: 30min

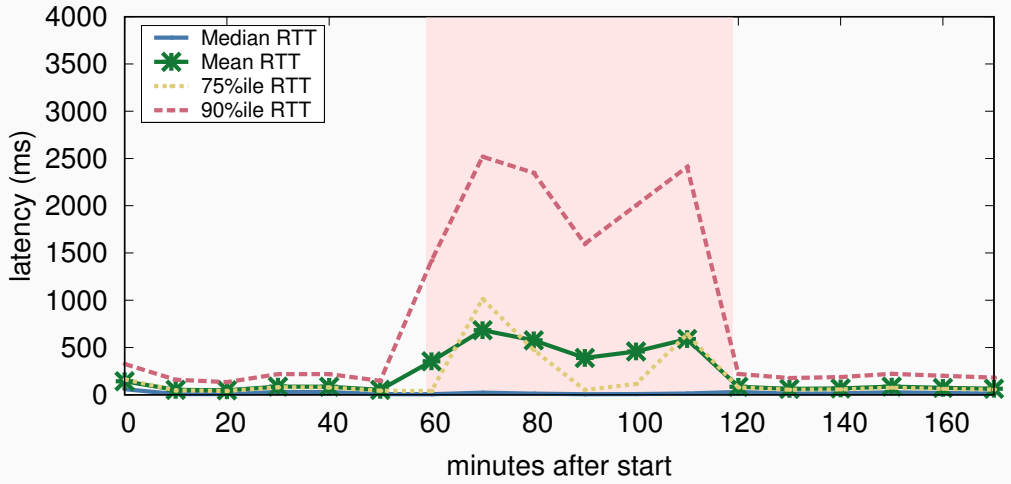




Good! Most clients are happy, as they retry (but takes longer)

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

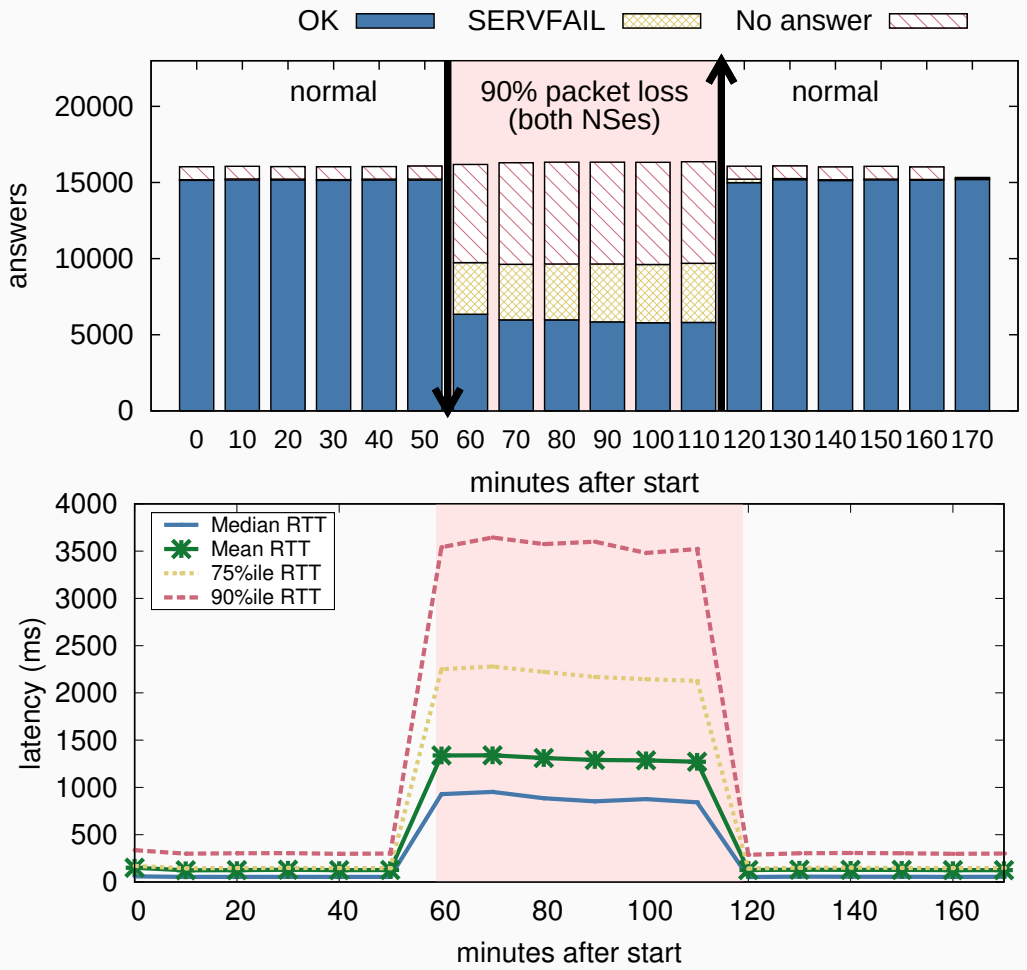




Good! Even at 90% packet loss with TTL 30min, most clients (60%) get an answer!! **Good Engineering!**

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

- What's TTL influence in partial DDoS?



Even with no caching (TTL 1min), 27% get an answer: stale + retries

Retries cost: hammering Auth servers

- Part of DNS resilience is that recursives keep on retrying
- There's a cost to it however: **8.1x** in case of no caching!
- Implications: OPS: be ready for **friendly fire**
  - usually not noticed during DDoS
  - If you overprovision level is 10x, know that 8.1x is friendly fire

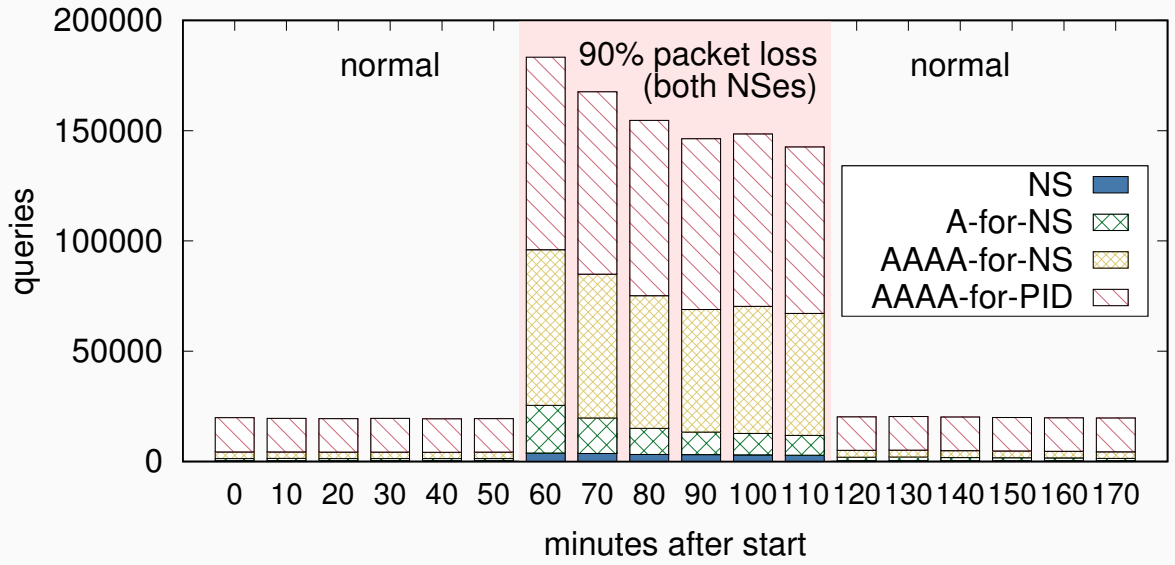


Figure 4: Queries received at Auth Servers .Experiment I: 90% success DDoS, TTL: 1min

Implications

- Caching and retries work *really well*
  - provided some authoritative stays partially up
  - and caches last longer than DDoS (as in TLDs, not in CDNs)
  - For DNS OPs: make one auth very strong? (careful with load distrubtion, see [5])
- Explains prior root DDoS outcomes

Implications

- There is a clear **trade-off** between TTL and DNS resilience
  - provided caches are filled and not about to expire
- Many commercial websites have short TTLs
  - explains the pain of Dyn‘s customers and users perception
  - shorter TTLs given them quicker management options (Amazon EC2 resolvers cap all answer TTL to 60s [7])

Conclusions

- First study to evaluate DNS resilience to DDoS from user's perspective
- Evaluate design choices of various vendors using measurements
- **Caching and retries**: important part of DNS resilience
  - Good engineering: thanks for all IETFers/devs who have built this
- Experiments show when they help and when they won't
- Consistent with recent outcomes
- DNS community:
  - There's a clear trade-off between TTL and DDoS robustness, choose wisely
  - Serving stale content is controversial, some deploy it

Discussion within DNS community

How do you set your TTL of your records?

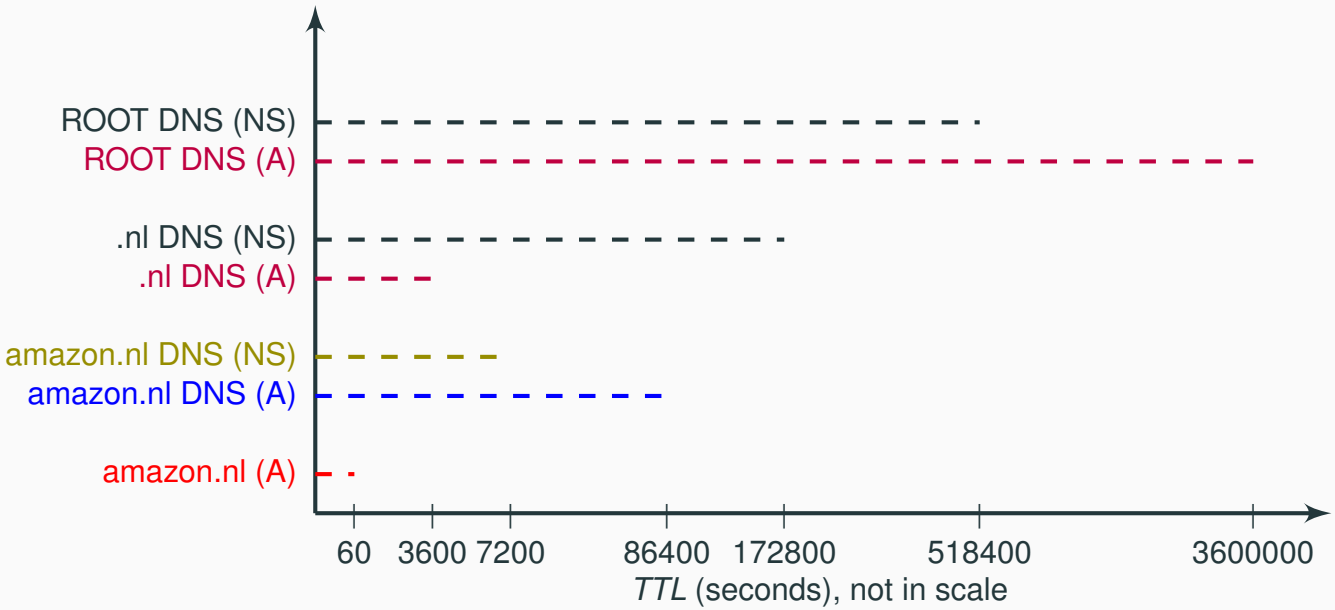


Figure 5: TTL relationships for `amazon.nl` on 20180813 – only showing authoritative data

Questions?

- Paper: https://www.isi.edu/~johnh/PAPERS/Moura18b.pdf
- Contact: giovane.moura@sidn.nl
- Thanks RIPE NCC and reviewers of various drafts:
  - Wes Hardaker, Duanne Wessels, Warren Kumari, Stephane Bortzmeyer, Maarten Aertsen, Paul Hoffman, our shepherd Mark Allman, and the anonymous IMC reviewers

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

Recursives in the wild: Engineering authoritative DNS servers.

In *Proceedings of the ACM Internet Measurement Conference*, pages 489–495, London, UK, 2017.

References iv

[6] Nicole Perlroth.

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

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

[7] Alec Peterson.

Ec2 resolver changing ttl on dns answers?

Post on the DNS-OARC dns-operations mailing list, https://lists.dns-oarc.net/pipermail/dns-operations/2017-November/017043.html, November 2017.

References v

[8] José Jair Santanna, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville, and Aiko Pras.

Booters—an analysis of DDoS-as-a-Service attacks.

In *Proceedings of the 14th IFIP/IEEE Interatinoal Symposium on Integrated Network Management*, Ottowa, Canada, May 2015. IFIP.

[9] 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.