WPI

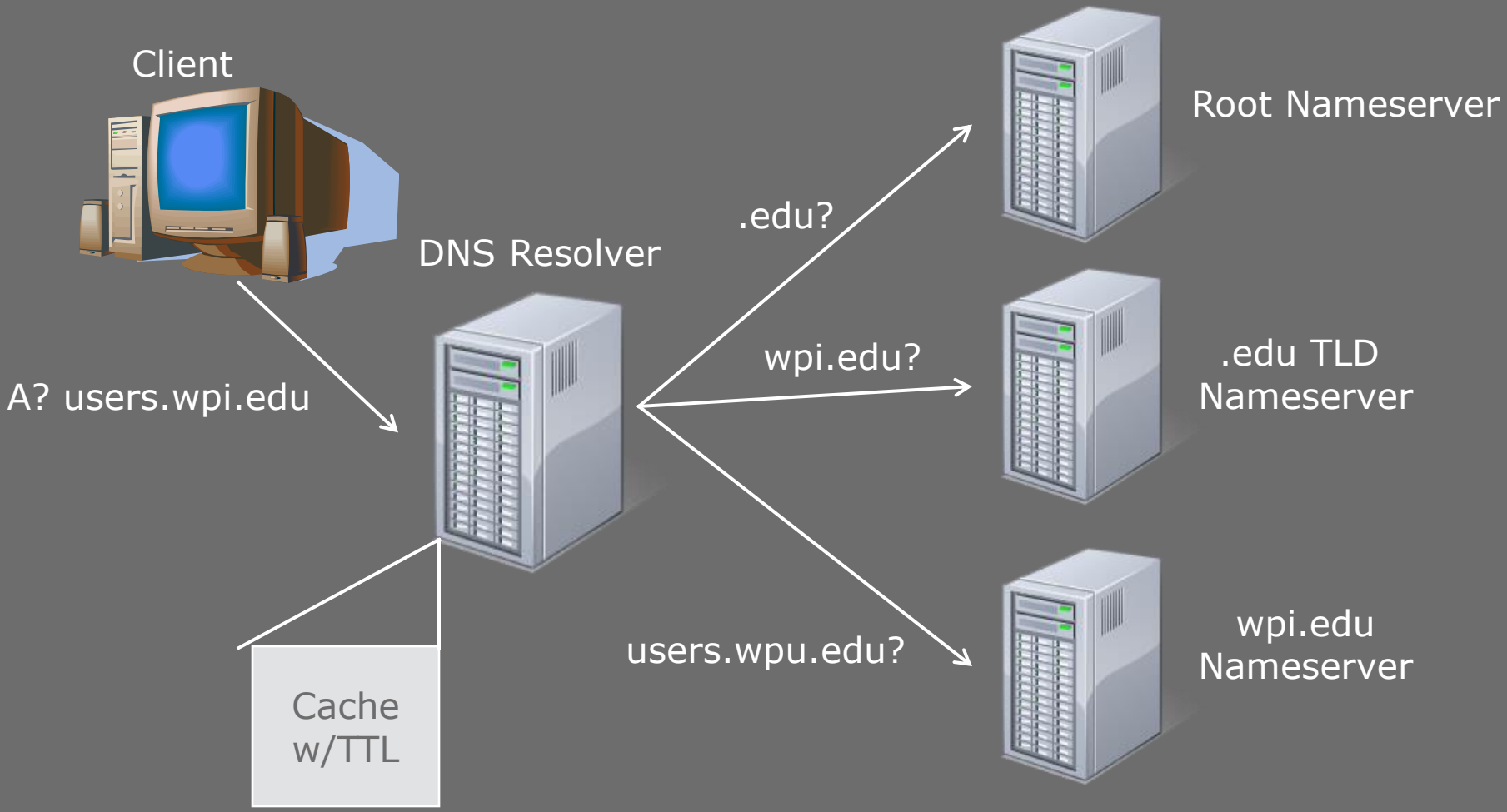
Mitigating DNS DoS Attacks

Hitesh Ballani, Paul Francis

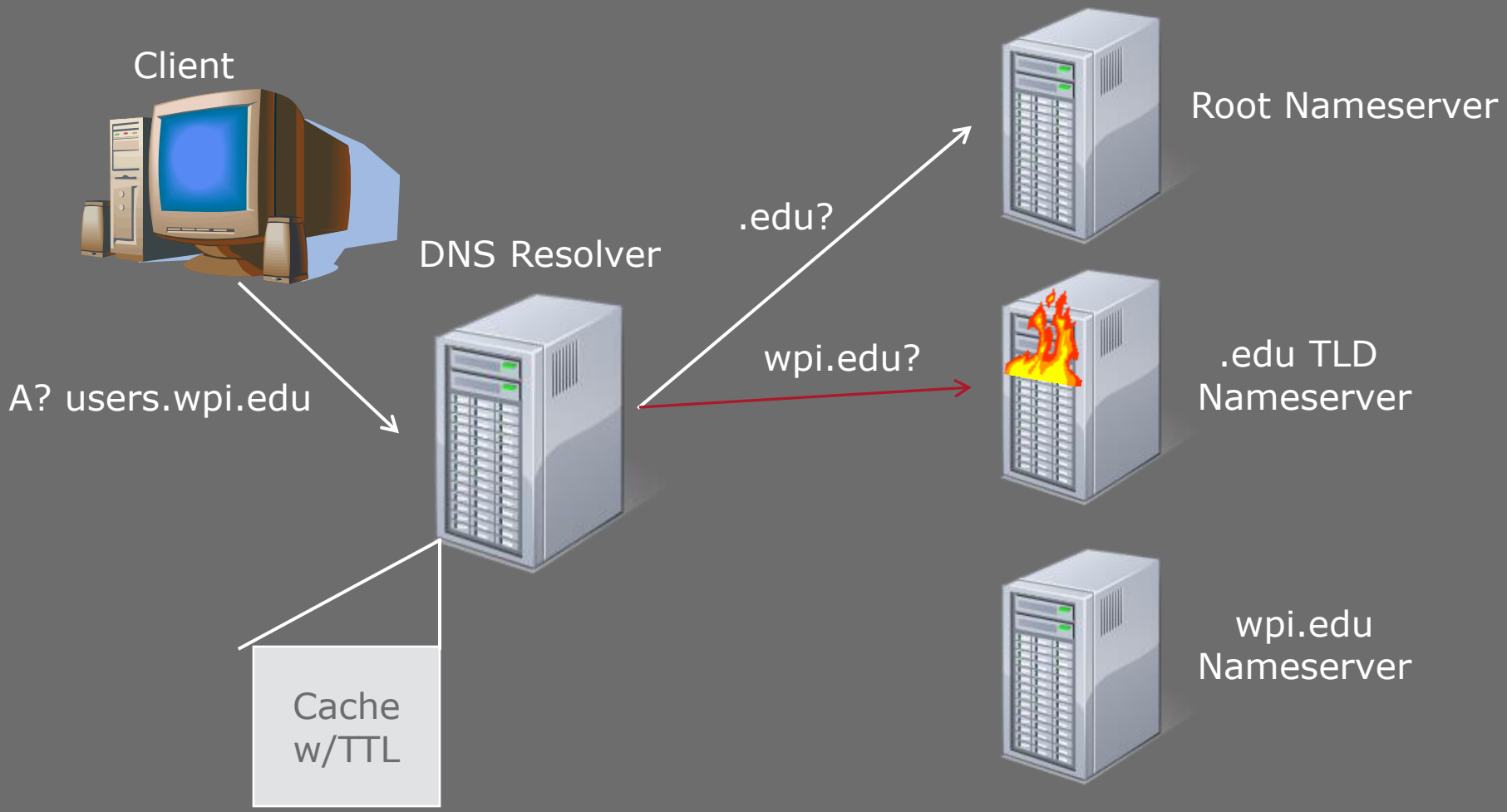
Outline

- What is DNS?
- What is the Problem?
- Solution
- Evaluation
- Conclusions

Domain Name System



Domain Name System - Problem



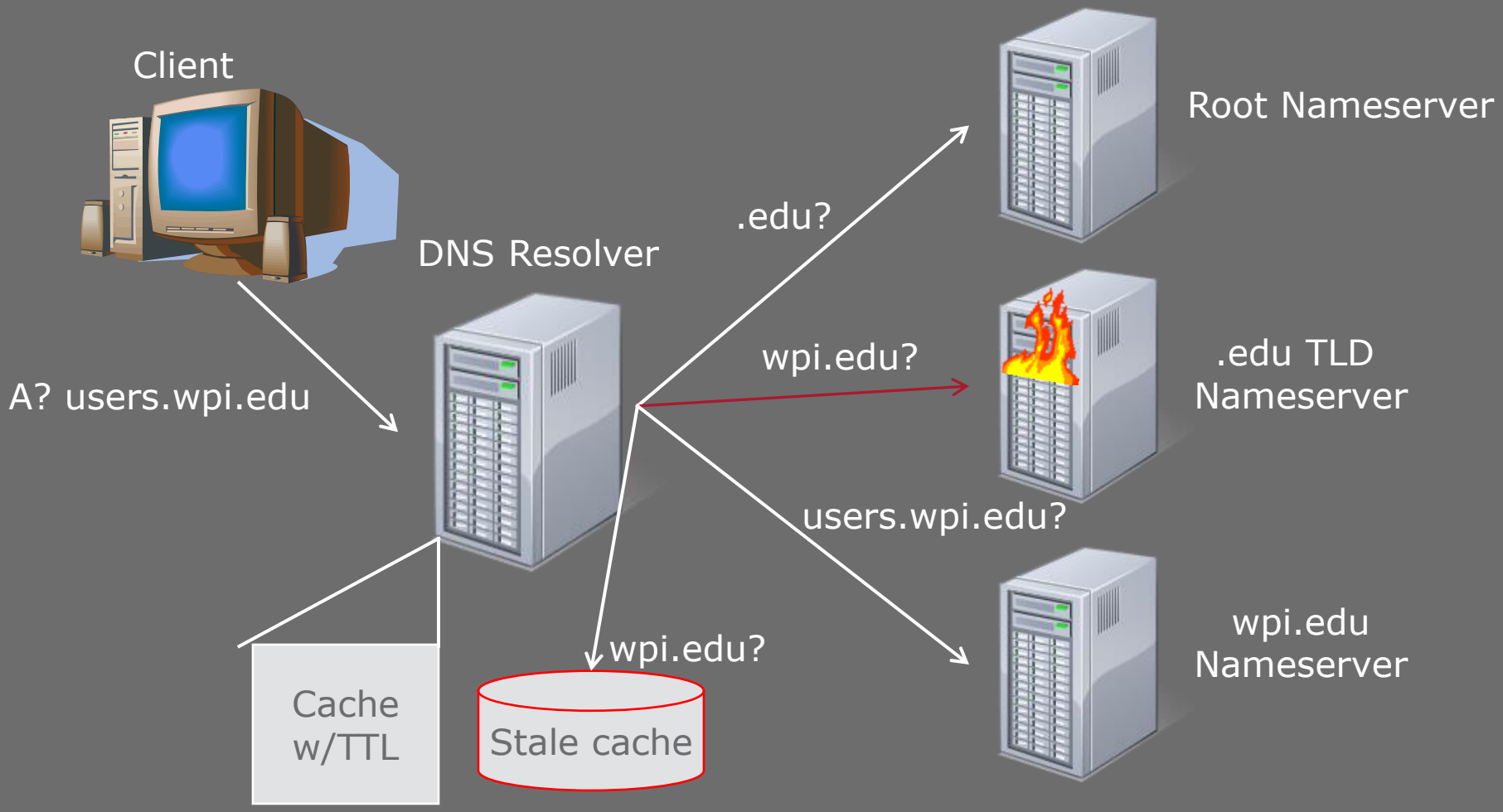
Domain Name System - Problem

- If the TTL has expired, a DNS resolver must rely on the availability of the authoritative nameservers.
- If any of the nameservers in the recurse chain are unavailable, resolution will fail.
- Thus, DoS attacks on nameservers can effectively ‘knock out’ a subset of the Internet.

Domain Name System - Solution

- The functionality to store old DNS lookups is already a component of DNS resolvers.
- Retain expired cached data beyond TTL in a ‘stale cache’.
- Use the stale cache as a ‘last resort’ if a step in the lookup chain is unresolvable.

Domain Name System - Solution



Additional Solution Parameters

- Whenever a nameserver is reached and a record updated, the stale cache must be cleared for that record.
- This guarantees that a stale cache record will always reflect the ‘last authoritative’ information our resolver saw.

Solution Evaluation

- Collected DNS traffic at the Cornell Computer Science link to the Internet
- 1300 hosts on the Cornell side of the link
- Sixty-five days of collected data (21 Nov 2007 – 24 Jan 2008
- 84,580,513 DNS queries/53,848,115 DNS responses
- This data was collected outside the first layer of resolvers, so client requests resolved by cache were not seen.

Solution Evaluation

- Two simulation parameters
  - Stale Cache Size: number of days beyond TTL to keep data in the stale cache (1-30 used)
  - Attack Duration: Length of availability attack on DNS nameservers (3,6,12,24 hours used)

Test 1 – No DNS Nameservers

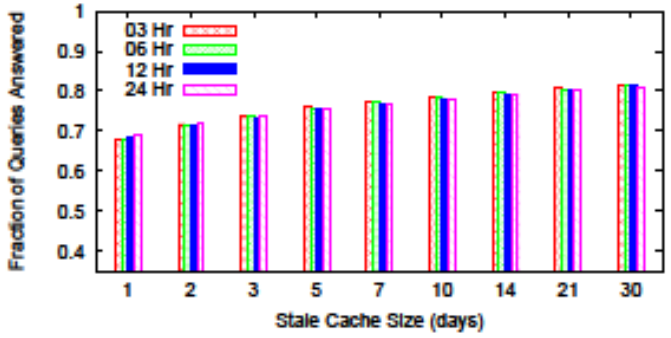


Figure 3: Fraction of Queries Answered using a stale cache of varying size during an attack wherein none of the nameservers are operational.

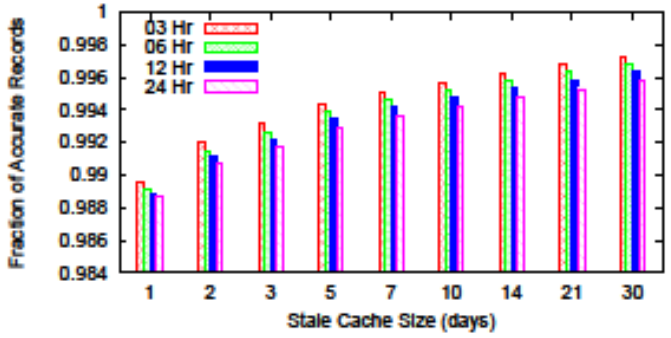
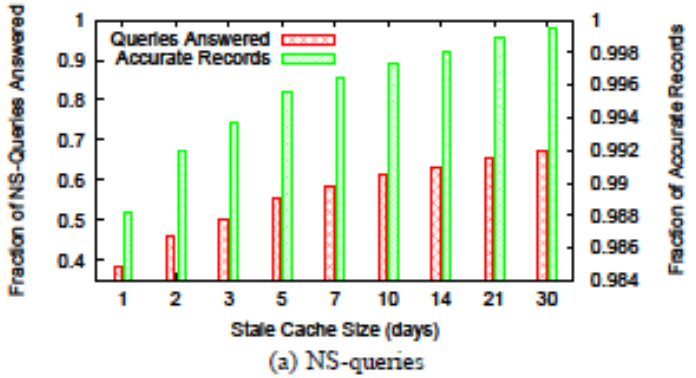
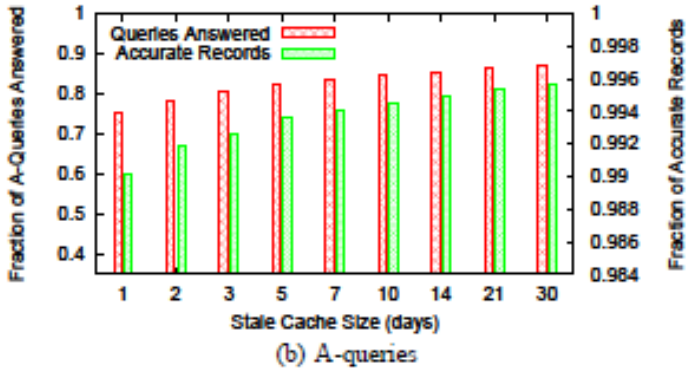


Figure 4: Fraction of Accurate Records in responses based on a stale cache of varying size during an attack wherein none of the nameservers are operational.



(a) NS-queries



(b) A-queries

Figure 5: For (a) NS-queries and (b) A-queries, Fraction of Queries Answered and Accurate Records when using a stale cache during an 3-hour attack.

From "Mitigating DNS DoS Attacks"

Test 2/3 – TLD/Second-level DoS

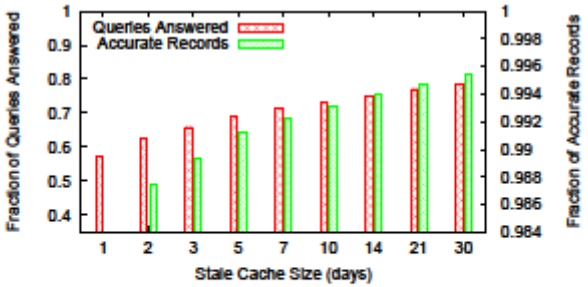


Figure 6: Fraction of Queries (for two-level names) Answered and Accurate Records when using a stale cache during an 3-hour attack on the TLD nameservers.

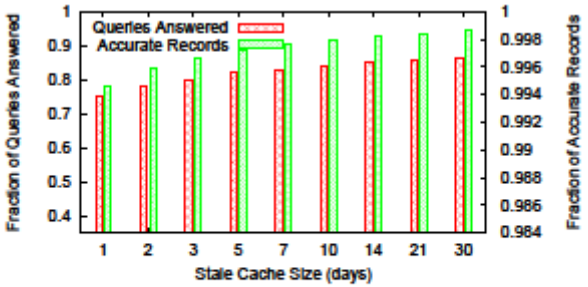


Figure 7: Fraction of Queries (for three-level names) Answered and Accurate Records when using a stale cache during an 3-hour attack on second-level nameservers.

Memory Footprint

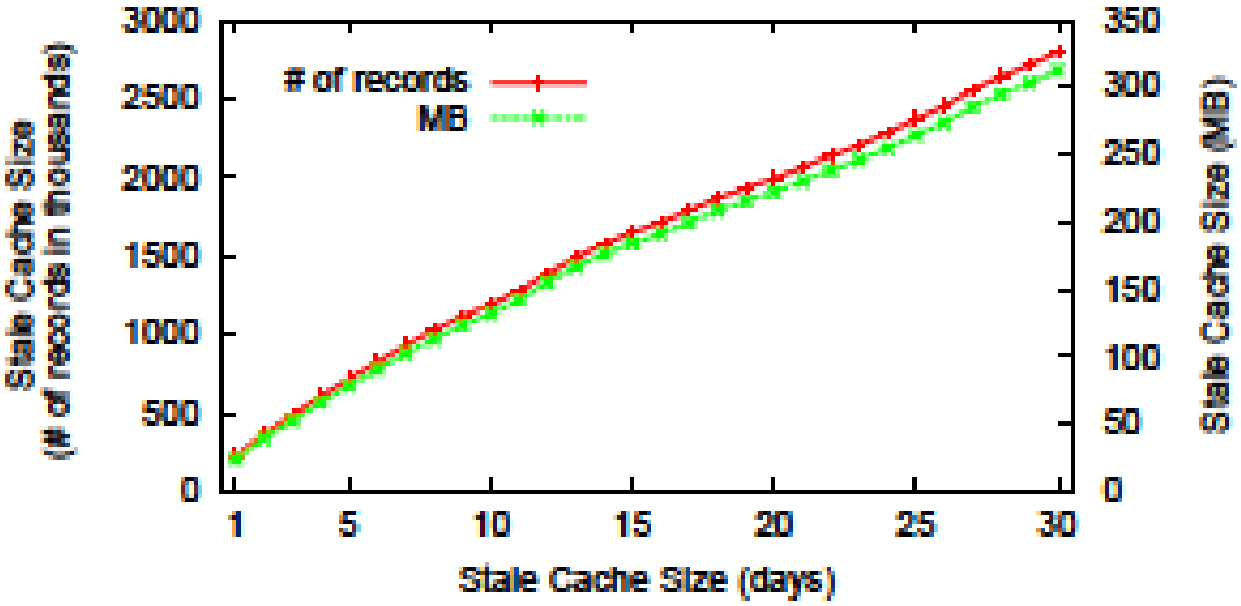


Figure 8: Stale cache memory footprint

Solution Evaluation - Pros

- The volume and accuracy of Stale Cache lookups demonstrates that this method can help DNS resilience in the face of a DoS attack.
- Does not impose additional DNS load.
- Does not affect query latency, since new lookup only occurs when ordinary lookup would have totally failed.
- Can be deployed incrementally.

Solution Evaluation – Cons

- Violates traditional DNS guarantee of accurate data outside of TTL period.
  - If DNS records have been changed since last access and outside of TTL, and
  - Zone nameservers are unavailable,
  - Stale Cache will return inaccurate DNS information.
    - Many DNS resolvers already disregard TTL, so server operators already have to prepare for incorrect resolutions, such as running duplicate servers for several weeks.
- Zone nameserver no longer has full control over sub-zone access.
  - Similar to above, a namserver may not be able to NXDOMAIN a sub-zone if it is unrechable outside of TTL

Solution Evaluation – Cons

- Both the above concerns rely on using the inaccessability of a Zone nameserver to cause the use of inaccurate data. This could be used in a timed attack by rogue subzones to force the inaccurate data to be used.

Conclusions

- The employment of a ‘stale cache’ at the DNS resolver level could significantly reduce the impact of DoS attacks on DNS nameservers.
- The primary drawback is the introduction of the possibility of inaccurate records being used if a record is updated and becomes inaccessible between resolver queries.