The Design and Implementation of a Next Generation Name Service for the Internet

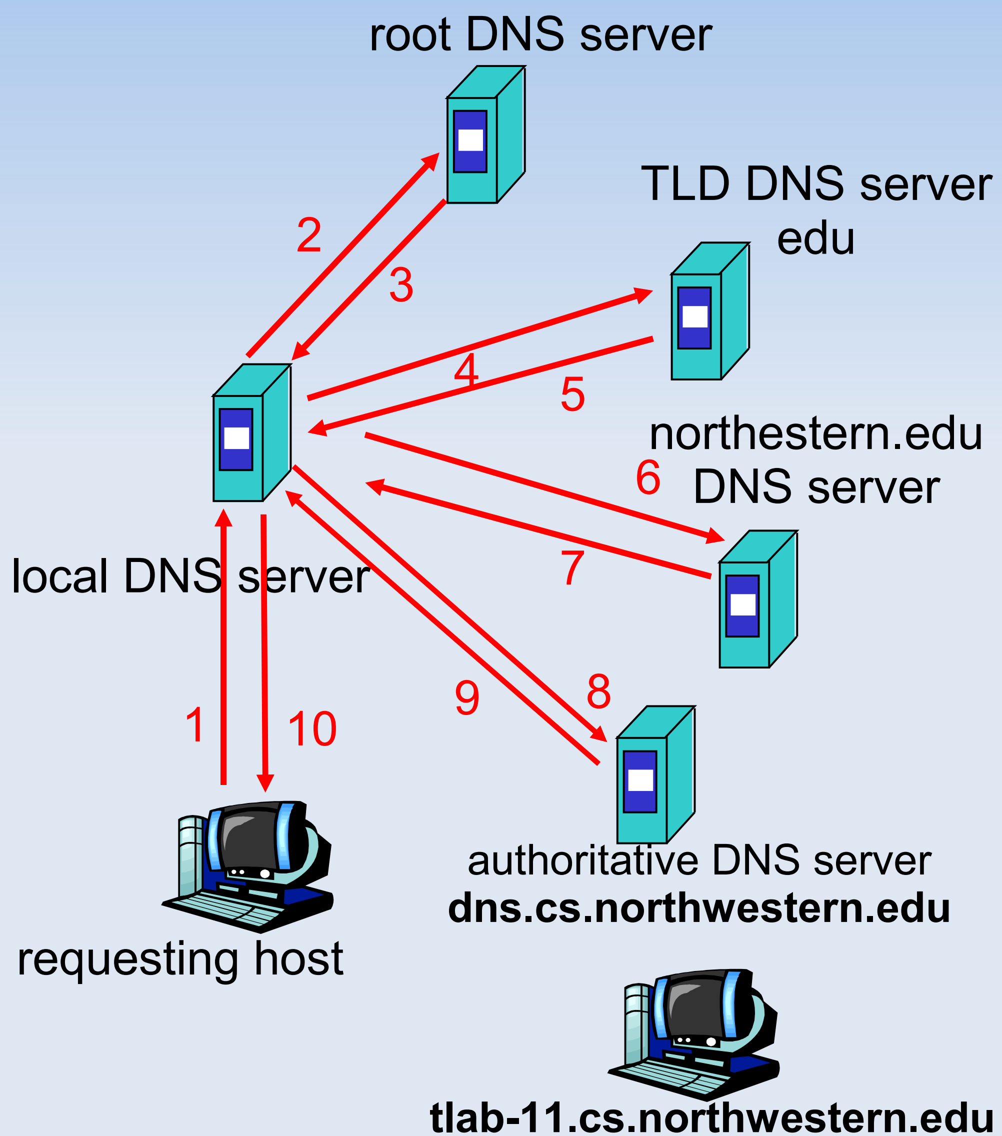
presented by Nikola Borisov

Outline

- Introduction
- DNS
- CoDoNS – Design and Implementation
- Performance comparison

DNS

- How does it work
  - Static distributed tree
  - Hierarchically partitioned NS
  - Non overlaping Domains
  - Delegating resposibility
  - 13 statical Ips
  - Resolvers



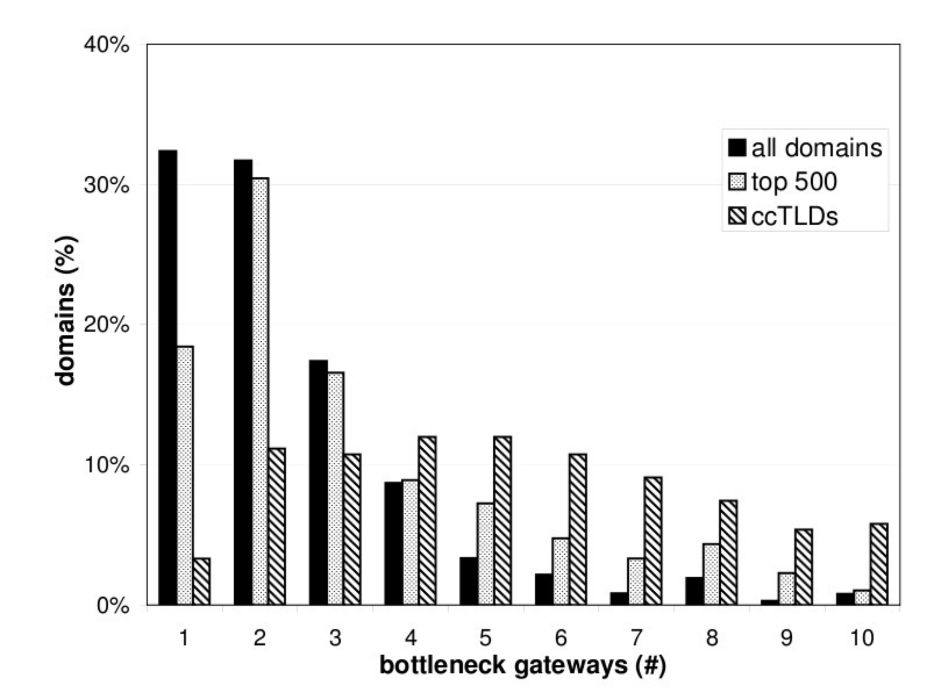
DNS Failure Resistance - Bottlenecks

- Experiment
  - 593160 unique domain names
  - 535088 unique domains
  - served by 164089 NS
- Most domains served by just 2 NS
  - DoS

Bottlenecks	All Domains	Top 500
1	0.82%	0.80%
2	78.44%	62.80%
3	9.96%	13.20%
4	4.64%	13.00%
5	1.43%	6.40%
13	4.12%	0.00%

DNS Failure Resistance – Network Bottlenecks

- Experiment
  - 10000 NS
  - 5000 domain names
  - PlanetLab traceroutes
- Results
  - 33% domains have a single gateway or router bottleneck



DNS Failure Resistance – Implementation Errors

- Survey 150000 NS for well known vulnerabilities
- 18% don't report versions
- 14% don't report valid versions
- 2% have *tsig* bug.
- 18% have *negcache* bug

DNS Performance – Latency

- Name resolution significan time consumer
  - 1 sec slow on 20% of web objects
  - 29% of queries take longer then 2 sec
- Low cache hit rates
- Dynamic server selection – short TTL
  - Creates big load on DNS servers
  - TTL < 15 min => significant cache hit rates drop

DNS Performance - Misconfiguration

- Broken or inconsistent delegations
  - 1.1% of resolution fail
  - 14% of authoritative NS return inconsistent responses
- Human errors in administration

DNS Performance – Load Imbalance

- DoS attacks friquent on Root and TLD
- Upper levels get more load

DNS Update Propagation

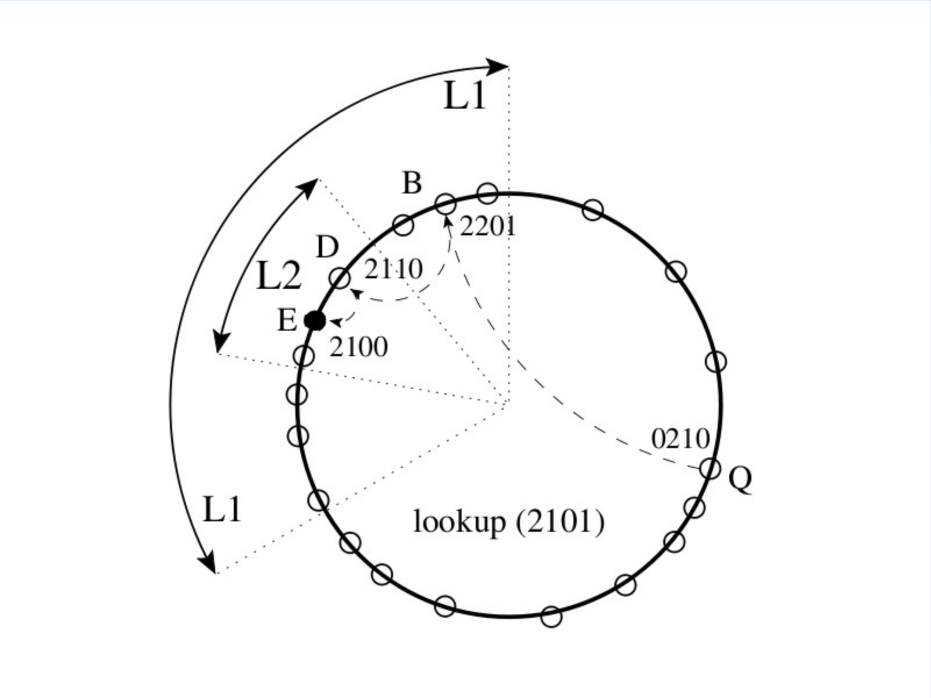
- Slow
  - 40% of domains have TTL > 1 day
  - Decreasing TTL increases cache misses
- Relocating resources

Cooperative Domain Name System

- Goals
  - Low latency
  - Resistant to DoS
  - Fast update propagation
- Overview
  - DHT based
  - Proactive caching layer – Beehive
  - DSN compatible

CoDoNS – How does it work?

- Prefix-matching DHT
  - Pastry, Tapestry
- O(logN) hops when routing
- Beehive caching
  - Replicate objects all nodes *i* matching prefixes
  - Vary *i*



CoDoNS (cont.)

- Vary replication to get desired latency
- Dinamicaly done by CoDoNS
- Popularity rank
  - local measurement
  - aggregation
  - determines the replication level

CoDoNS – Replication

- Push like protocol
- Recursive
  - push only to nodes with one prefix less
  - replicate further
- Fast updates
- Joining nodes
  - miss update
  - performance penalty but no stale data

CoDoNS - Architecture

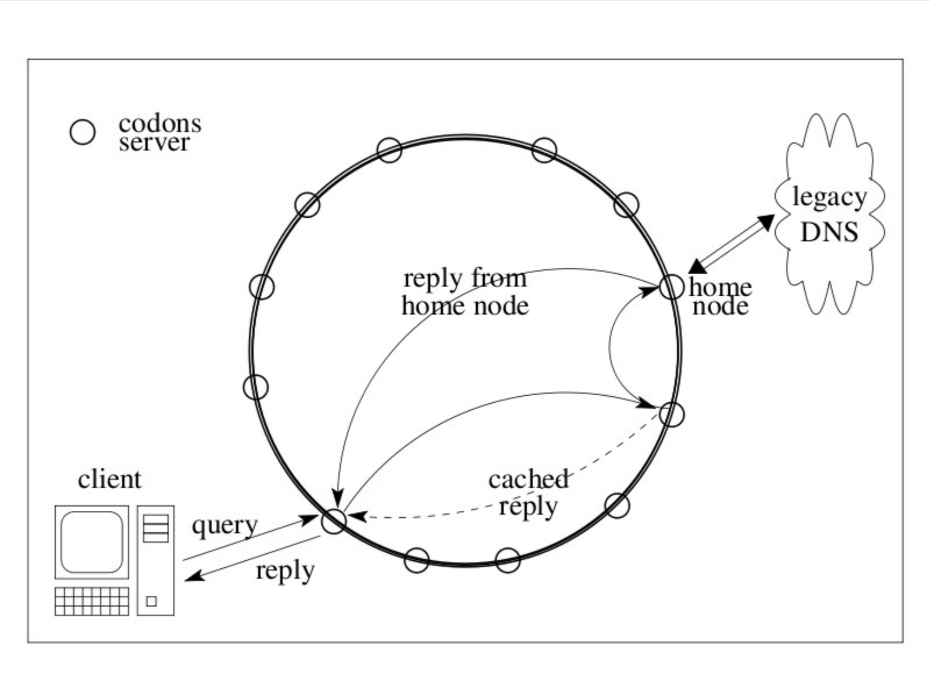
- Globaly distributed
- Peer-to-Peer
- Each institution contributes machines
- DSN compatible
  - no client changes required
- Decouples namespace management from query resolution
  - Nameowners purchase certificates

CoDoNS – Architecture (cont.)

- No restrictions on names
- insert, delete, update
- Avoiding data loss with replication

DNS to CoDoNS transition

- Home node queries DNS
- Home node caches result
- Direct Caching
- Proactively refetches legacy DNS records
- Small TTL redirection
- NXDOMAIN



CoDoNS Issues and Implications

- DNSSEC – authentications of records
  - Namespace operator
    - signs records
    - upper level domains can verufy the signiture
  - Clients
    - can verify records
- CoDoNS caches certificates
- Non DNSSEC clients trust only local CoDoNS
- Certificated needed for insert, update, delete

CoDoNS Issues and Implication 2

- Namespaces can be co-managed
- DNSSEC not used by all DNS servers
- Malicious nodes
  - secure routing table
  - increased lookup latency
- Dynamic name resolution
  - redirection record

CoDoNS Evaluation

- Setup
  - PlanetLab
  - Compare CoDoNS and lagacy DNS
  - 281 943 queries for 47230 domains
  - 75 geographicaly distributed nodes

Lookup Performance

- 50% answered immediately
- median 2ms compared to 39ms

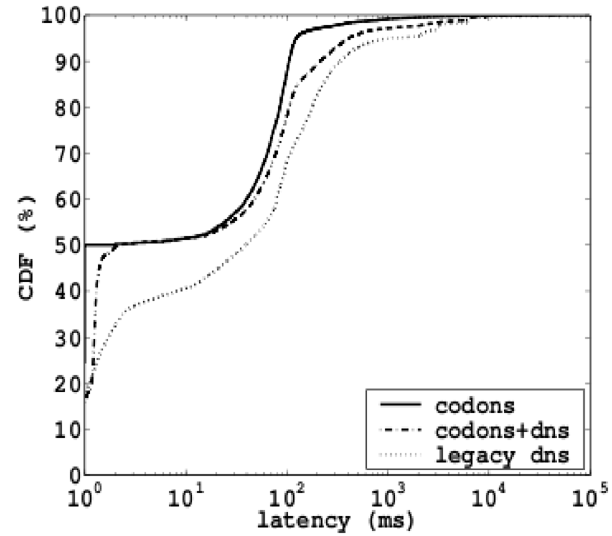


Figure 5: Cumulative Distribution of Latency: CoDoNS achieves low latencies for name resolution. More than 50% of queries incur no network delay as they are answered from the local CoDoNS cache.

Latency	Mean	Median	90^{th} %
CoDoNS	106 ms	1 ms	105 ms
CoDoNS+DNS	199 ms	2 ms	213 ms
Legacy DNS	382 ms	39 ms	337 ms
PlanetLab RTT	121 ms	82 ms	202 ms

Table 4: Query Resolution Latency: CoDoNS provides low latency name resolution through analytically informed proactive caching.

Lookup Performance

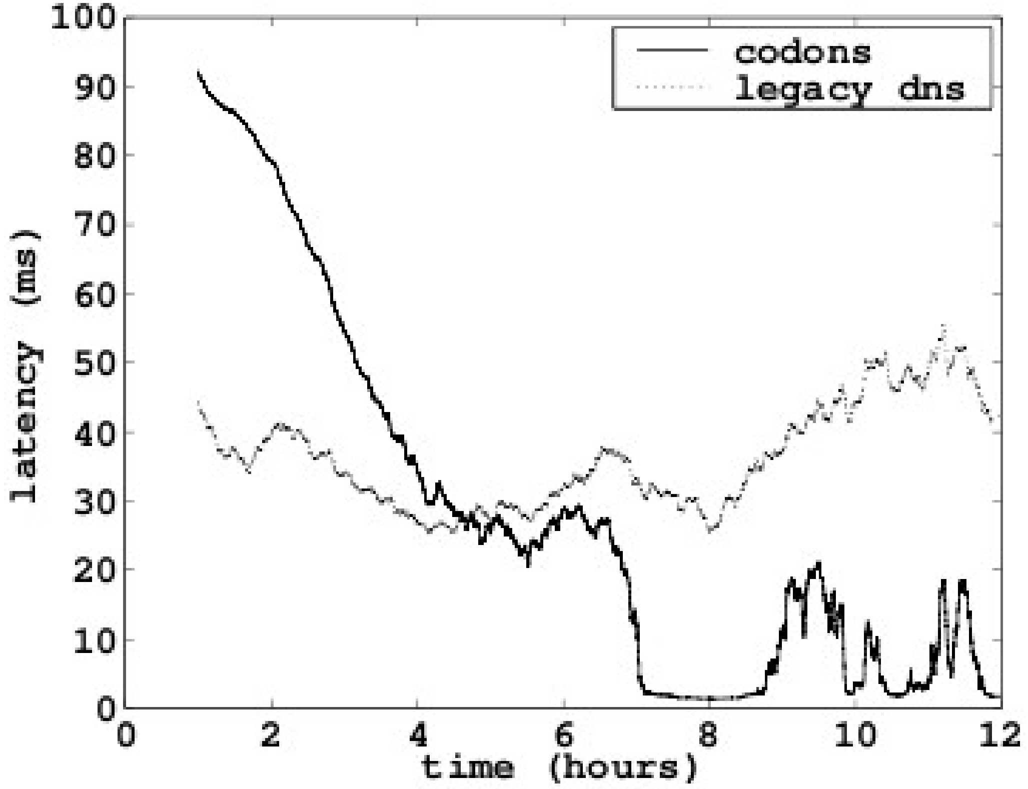


Figure 6: Median Latency vs Time: Lookup latency of CoDoNS decreases significantly as proactive caching takes effect in the background.

Flash-crowd Effect

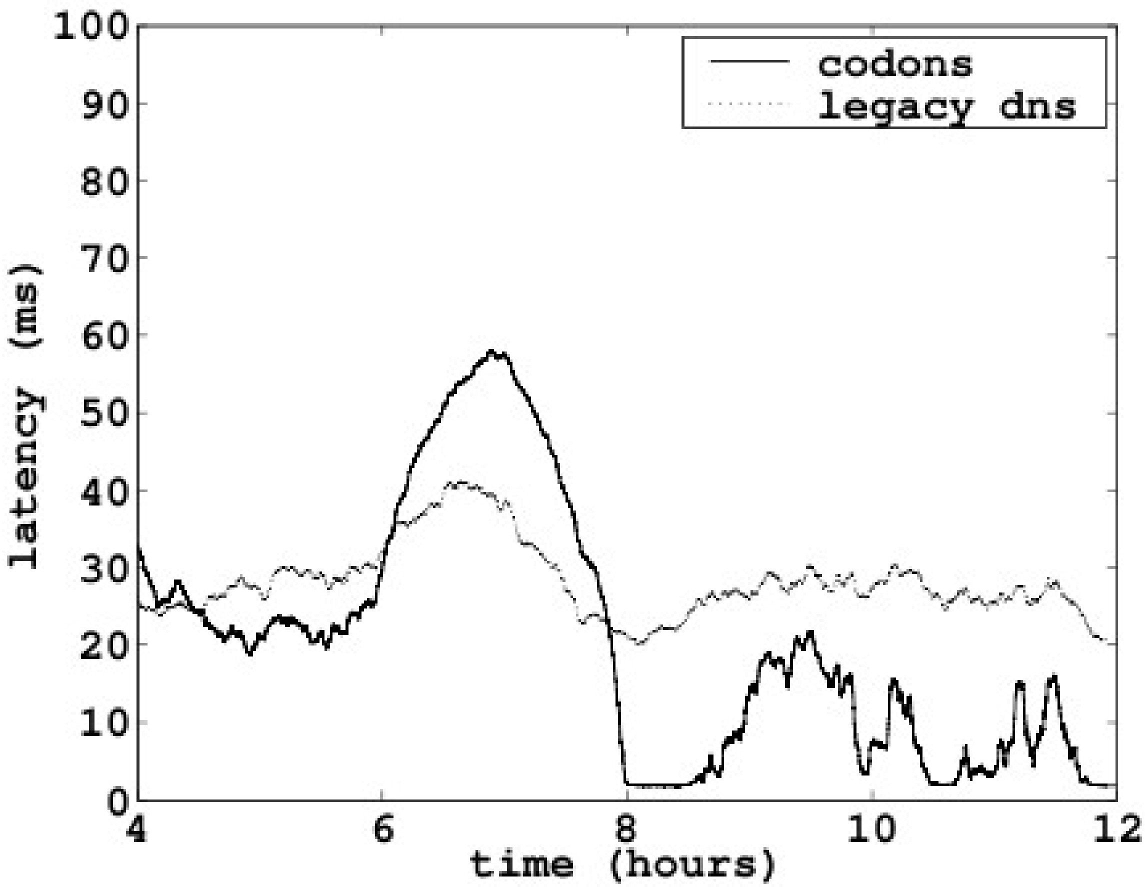


Figure 7: Median Latency vs Time as a flash-crowd is introduced at 6 hours: CoDoNS detects the flash-crowd quickly and adapts the amount of caching to counter it, while continuing to provide high performance.

Load Balance

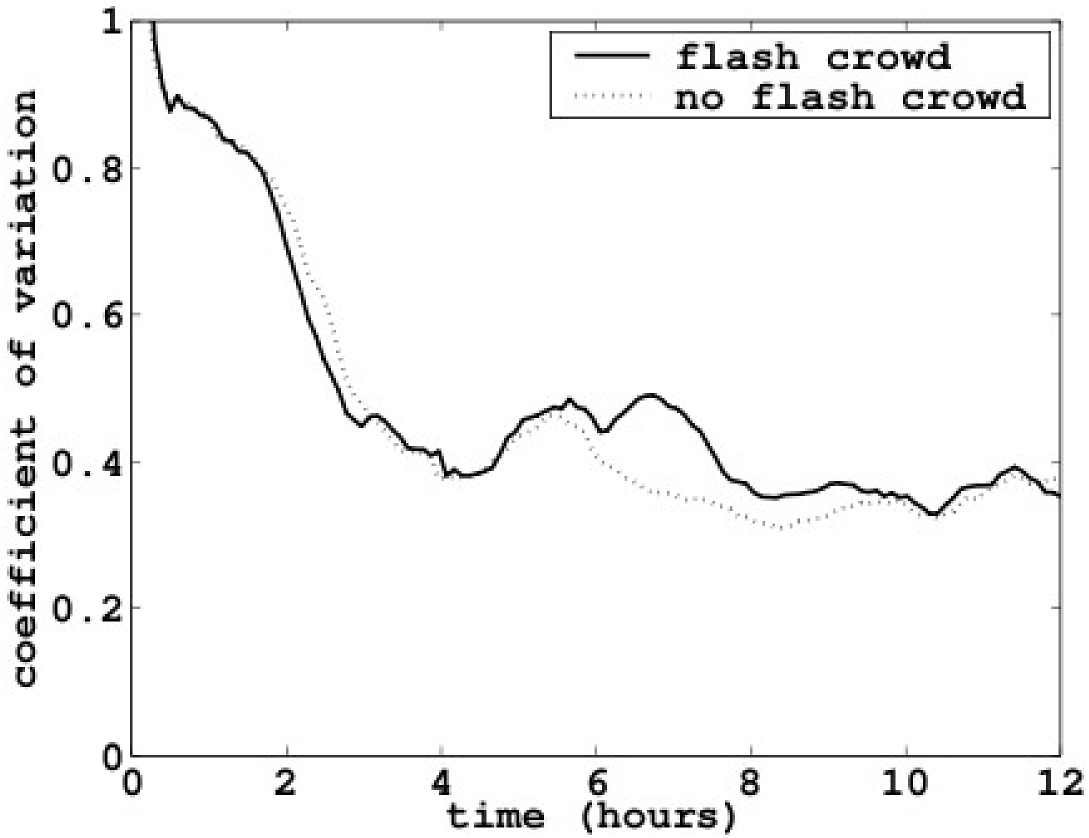


Figure 8: Load Balance vs Time: CoDoNS handles flash-crowds by balancing the query load uniformly across nodes. The graph shows load balance as a ratio of the standard deviation to the mean across all nodes.

Update Propagation

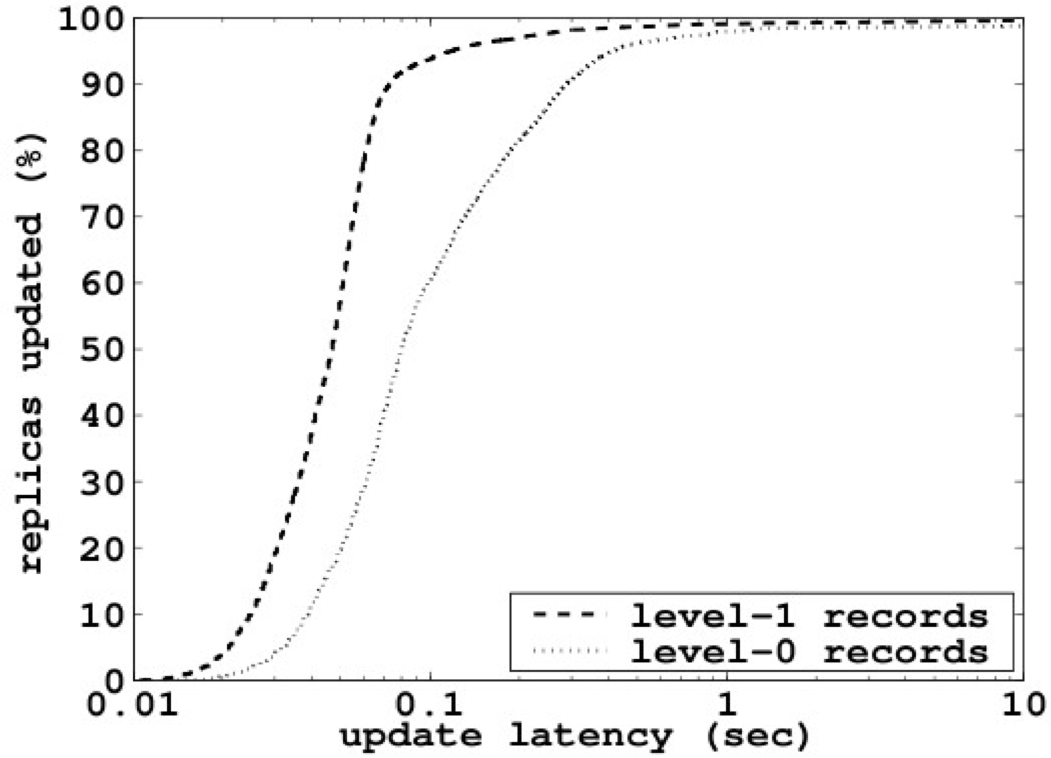


Figure 9: Update Propagation Time: CoDoNS incurs low latencies for propagating updates. 98% of replicas get updated within one second.