Strong Cache Consistency Support for Domain Name System

2004 SIGCOMM Poster

Xin Chen, Haining Wang, Shansi Ren and Xiaodong Zhang
College of William and Mary, Williamsburg, Virginia

For more information, contact:
Xin Chen,
Department of Computer Science
College of William and Mary
P.O. Box 8795
Williamsburg, VA 23185
Tel: 757 221-3477
E-mail: xinchen@cs.wm.edu

Motivation

TTL-Based Cache Consistence:
- Originally designed for static domain name mapping
- Only weak consistency provided

Current DNS Cache Updates:
- Set a short TTL before update (2-3 days)
- Resume to a normal TTL after update (2-3 days)
- Long update delays even changes are anticipated!

Problems: (in the changing world!)
- Unpredictable mapping changes: many changes are unexpected while critical services need always-on availability
- Dynamic domain name mapping: widely deployed dynamic DNS solution sets up servers on temporal IPs from DHCP
- Emergence events to support: Web servers are closed/moved at emergence (e.g. 911, nature disaster, etc.)
- Redundant DNS traffic: Content Delivery Network providers use small TTLs to achieve load balance among their surrogates

Objective

An effective solution for DNS cache consistency!

DNS Dynamics Measurement

❖ How often does a domain name to IP address mapping change?

SOA: *authority indication for a zone;*
A: *hostnames to IP address mappings;*
PTR: *IP addresses to hostname mappings;*
NS: *domain name server reference lists for a zone;*
MX: *mail exchangers for a domain.*

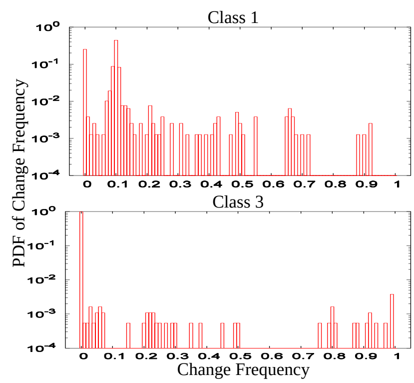
- DNS resource records are changed for different purposes
  - 'A' records -- most used, have significant effects if changed
  - our measurements are focused on 'A' records

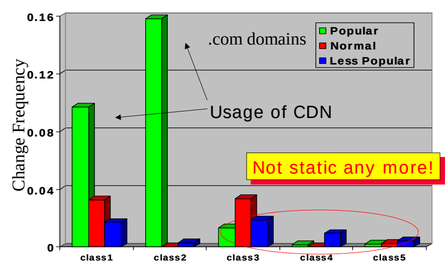
Methods

- Domain Name Collection
  - IRcache: Nov. 5 – Nov. 11, 2003
- Domain Name Classification
  - TLDs: .com, .net, .org, .edu, cc domains
  - CDNs: identified by specific strings of CDN providers
  - Dyns: identified by specific strings of dynamic DNS providers
  - 5 classes: based on domains' TTLs
- Measurement Period
  - Nov. 30, 2003 – Jan. 3, 2004

Class	TTL	Resolution	Duration	Domain number
1	[0,1m)	20 sec	1 day	803
2	[1m, 5m)	1 min	3 days	934
3	[5m,1h)	5 min	3 days	2020
4	[1h,1d)	1 hour	7 days	7217
5	[1d,inf)	1 day	1 month	4473

Measurement Results



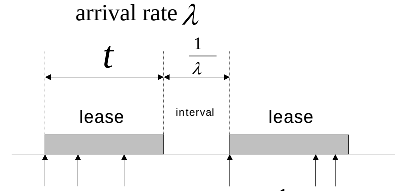


Dynamic Lease

Lease: a combination of polling and invalidation

Challenge: lease length selection
- long leases: more storage overhead
- short leases: more network traffic

Assumption: request intervals follow Poisson distribution with average arrival rate λ



Storage overhead: $P = t/(t+\frac{1}{\lambda})$

Communication overhead: $M = 1/(t+\frac{1}{\lambda})$

Problem Definition:

Storage-constrained lease: minimize the communication overhead given the storage allowance

Analysis: equivalent to a Knapsack problem

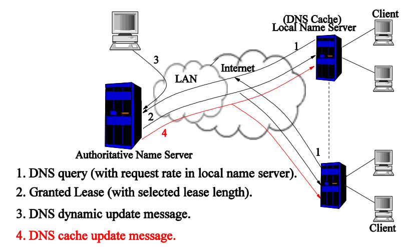
Optimal solution: maximal lease length granted to the caches with the highest query rate (dynamic lease), because: $\boxed{\frac{\Delta M}{\Delta P} = \lambda}$

Communication-constrained lease can be defined and solved in a similar way.

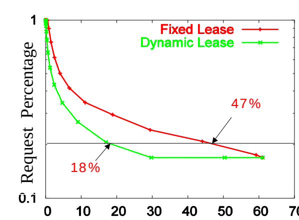
Our Solution — DNScup

DNS Cache Update Protocol

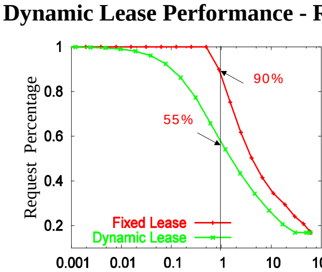
Basic idea: an authoritative name server uses dynamic lease technique to notify relevant caches when its resource record changes.



Dynamic Lease Performance - Storage

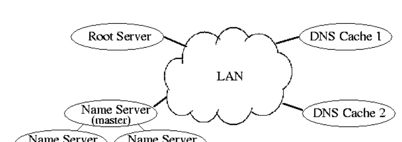


Dynamic Lease Performance - Request



Implementation

- Efficiency
  - UDP: first choice
  - Update propagation without NOTIFY
- Robustness
  - Name server repeats sending until ACK received
  - DNS cache validates all records after reboot
- Compatibility
  - Name server supports both TTL and DNScup mechanisms
  - DNS cache can use both TTL and lease
- Security
  - Name server uses TSIG to control updates
  - DNS cache uses ACK to verify updates



Implementation test bed