Reducing DNS Caching or “How low can we go?”

Saleem Bhatti
School of Computer Science
University of St Andrews

Randall Atkinson
Cheltenham Research
Motivation for examining DNS

<table>
<thead>
<tr>
<th>Layer</th>
<th>IP</th>
<th>ILNP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>FQDN or IP address</td>
<td>FQDN</td>
</tr>
<tr>
<td>Transport</td>
<td>IP address + port no.</td>
<td>Identifier + port no.</td>
</tr>
<tr>
<td>Network</td>
<td>IP address</td>
<td>Locator</td>
</tr>
</tbody>
</table>

- ILNP: cleaner semantics for naming in the IP stack:
  - Assumes consistent use of DNS names in applications
  - Uses DNS for mobility, multi-homing, traffic engineering ...

- So, DNS performance is important for ILNP
- **But this talk is not about ILNP ...**
Domain Name Service

• Distributed name resolution service
• Maps Fully Qualified Domain Names (FQDNs) to DNS records:
  e.g. FQDN (www.cs.st-andrews.ac.uk) to a DNS A record (IPv4 Address, 138.251.206.45),
• Also provides other administrative data for specific services (a simple directory service):
  – Name Servers for a domain (NS records)
  – Mail servers (MX records)
  – Jabber (and other) servers (SRV records)
  – Other record types are possible ...
DNS System Architecture

• Globally distributed name space
• Globally distributed name servers each holding mappings for part of the name space
• Traditionally, read-only for end users
• Enhancements now widely available to enable write access for end users:
  – Secure DNS Dynamic Update (RFC-3007)
  – DNS Security (RFC-4033 to 4035) also useful
  – Implemented in BIND, MS Windows, MS Server
DNS Lookup Sequence

1. "A" query for www.bar.com
2. "NS" query for .COM
3. "NS" query for BAR.COM
4. "A" query for WWW.BAR.COM
6. Data using addr for www.bar.com
Dynamic write access to DNS

• **Write access** is now available for end users
• There is a **temporal caching hierarchy** across the **spatial** distribution of names:
  – Different **records** get cached for different periods of **time**, e.g. NS records and A records
  – Maximum caching time defined by a **Time To Live (TTL)** value held in each DNS record.

• **Could these two features be exploited in some sensible ways?**
(Non-)Effectiveness of DNS caching


• DNS caching is ineffective for edge sites:
  – trace-driven emulation (no experiments)
  – A records could have low TTL (e.g. below 1000s)
  – such low TTL would have low impact on DNS load
DNS experiments at StA [1]

• Experiments in Q4/2009
• Modify TTL values of records in operational DNS server at School of CS, St Andrews
  – 4 DNS servers: Windows ActiveDirectory
  – ~400 DNS clients: Windows, Linux, MacOSX, BSD
• TTL values for successive 7-day periods during normal semester:
  – changed DNS TTL on ActiveDirectory
  – used TTL values 1800s, 30s, 15s, 0s
• Configured clients not to cache.
DNS experiments at StA [2]

- Passive collection of packets via port mirror:
  - `tcpdump(8)` targeting `port 53`
  - Captured all DNS packets
- Results shown on following slides are for:
  - A record requests for servers only during the capture period (relevant to ILNP, and less ‘noisy’ data)
  - using 1 second buckets
- Basic statistics:
  - on time-domain data
- Spectral analysis:
  - examination of request rates
- Analysis: home-brew `python` scripts, NumPy package
## 2009: Basic dataset meta-data

( awaiting verification )

<table>
<thead>
<tr>
<th>Data set name</th>
<th>TTL [s]</th>
<th>Duration [s]</th>
<th>Total DNS packets captured</th>
<th>Number of A record requests for 67 servers</th>
</tr>
</thead>
<tbody>
<tr>
<td>dns1800</td>
<td>1800</td>
<td>601,200</td>
<td>41,868,522</td>
<td>2,004,133</td>
</tr>
<tr>
<td>dns30</td>
<td>30</td>
<td>601,200</td>
<td>71,105,247</td>
<td>2,648,796</td>
</tr>
<tr>
<td>dn15</td>
<td>15</td>
<td>601,200</td>
<td>56,472,027</td>
<td>3,240,675</td>
</tr>
<tr>
<td>dns0</td>
<td>0</td>
<td>601,200</td>
<td>55,868,573</td>
<td>4,501,590</td>
</tr>
</tbody>
</table>

---

1 from tcpdump timestamps, rounded to nearest second, 7 days = 604,800 seconds, less 3600s temporal guard band for TTL value changes = 601,200 seconds

2 includes all request and response packets to/from port 53 (TCP and UDP), including erroneous requests etc

3 servers that were active during the 4 weeks of data capture
dns1800: A record requests TTL=1800s

Mean: 3.33 request/s
Std Dev: 3.47 requests/s
Max: 183 requests/s
dns30: A record requests TTL=30s

Mean: 4.41 request/s
Std Dev: 4.27 requests/s
Max: 261 requests/s
dns15: A record requests TTL=15s

Mean: 5.39 request/s
Std Dev: 4.85 requests/s
Max: 123 requests/s
**dns0: A record requests TTL=0s**

![Graph showing DNS A record queries, dns2009-0000]

- **Mean:** 7.49 requests/s
- **Std Dev:** 4.93 requests/s
- **Max:** 3.69 requests/s
### 2009 Summary of basic statistics

(remaining verification)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>dns1800</td>
<td>3.33</td>
<td>3</td>
<td>3.47</td>
<td>183</td>
</tr>
<tr>
<td>dns30</td>
<td>4.41</td>
<td>4</td>
<td>4.27</td>
<td>261</td>
</tr>
<tr>
<td>dns15</td>
<td>5.39</td>
<td>4</td>
<td>4.85</td>
<td>123</td>
</tr>
<tr>
<td>dns0</td>
<td>7.49</td>
<td>7</td>
<td>4.93</td>
<td>369</td>
</tr>
</tbody>
</table>

**60x drop** in TTL values results in **⅓x increase** in A record requests:

**0 TTL** gives (only) **2¼x increase**.
2009 Basic spectral analysis

- Create periodogram by counting frequency of bucket sizes:
  - have used 1s bucket
  - so size of bucket is number of requests/s
- Comparison of periodogram:
  - shows changing dynamics of request rates
  - gives a better understanding of the trends in rates
2009 dns1800 perdiogram

7-day DNS A record query rates, dns2009-1800

7-day CDF for DNS A record query rates, dns2009-1800
2009 dns30, dns15, dns0 perdiograms

7-day DNS A record query rates, dns2009-0030

7-day CDF for DNS A record query rates, dns2009-0030

7-day DNS A record query rates, dns2009-0015

7-day CDF for DNS A record query rates, dns2009-0015

7-day DNS A record query rates, dns2009-0000

7-day CDF for DNS A record query rates, dns2009-0000

7-day DNS A record query rates, dns2009-0030

7-day CDF for DNS A record query rates, dns2009-0030
So far, our discussion presents a technical (academic) curiosity.

The interesting question really is: “So, what?”
What is possible if DNS TTL were zero?

• Dynamic, frequent, & authenticated DNS updates possible:
  – Simulated by Pappas, Hailes, & Giaffreda, published in LCS 2002
  – Very useful for mobility/multi-homing aspects of ILNP
• Help defend against certain network attacks:
  – DNS cache poisoning for end-sites (that do not use DNSSEC)
  – DDoS: fast-cycle multi-homing (i.e. a kind of “fast-flux” DNS for defence rather than attack)
  – Others possible …
• Support for mobility and multi-homing:
  – Location updates give changes in connectivity
• Potential for edge-site based multi-path and TE control:
  – multiple Locator values and DNS L record preferences
• High-speed load balancing and VM mgmt for data centres
Who would set DNS TTLs so low?

• Real A record values for some servers:
  – TTL = 60 seconds: www.yahoo.com
  – TTL = 20 seconds: content servers in akamai.net
  – TTL = 0 seconds: www.cs.st-andrews.ac.uk

• Note that a site would **NOT** set low TTLs for:
  – Its own NS records, which identify its DNS servers.
  – The A records related to its NS records.
  – A, CNAME, PTR records for services, e.g. email MX
  – A (mobile) site can make remote some or all of its authoritative DNS servers; some sites do so today.
Operational Considerations

• Implied semantics of TTL value:
  – gotcha: some systems assume that, if network outage time > TTL, then service is down

• PAM2003, PAM2004, NETTS2004 papers by Wessels et al.:
  – http://dns.measurement-factory.com/writings/

• In fact, the main site has some very interesting reading, including:
  – http://dns.measurement-factory.com/surveys/200810.html
Ongoing DNS experiments at StA

- More measurements in 2010/2011:
  - Secure Dynamic DNS updates
- Assess feasibility of supporting ILNP:
  - mobility
  - multi-homing
  - traffic engineering
- Q4/2009 and Q1/2008 data:
  - lots of interesting “features” and behaviour visible in DNS data
  - analysis in progress ...
Summary and Conclusion

• **Summary:**
  – Zero TTL values for edge-site DNS records possible
  – DNS load with zero DNS TTLs seems manageable

• **Conclusion:**
  – Both frequent DNS accesses and frequent Dynamic DNS Updates seem practical to deploy

• **A Very Big Thanks to:**
  – **Systems Group** at cs.st-andrews.ac.uk for implementing DNS TTL changes