University
of
St Andrews

Reducing DNS Caching

Saleem Bhatti
School of Computer Science
University of St Andrews

Reducing DNS Caching
Saleem N. Bhatti, Randall Atkinson
14th IEEE Global Internet Symposium (GI2011), Shanghai, 15 Apr 2011

1. **The DNS zero TTL challenge**
2. Experiment configuration
3. Observations and analyses
4. Round-up

Motivational example: mobility in ILNP

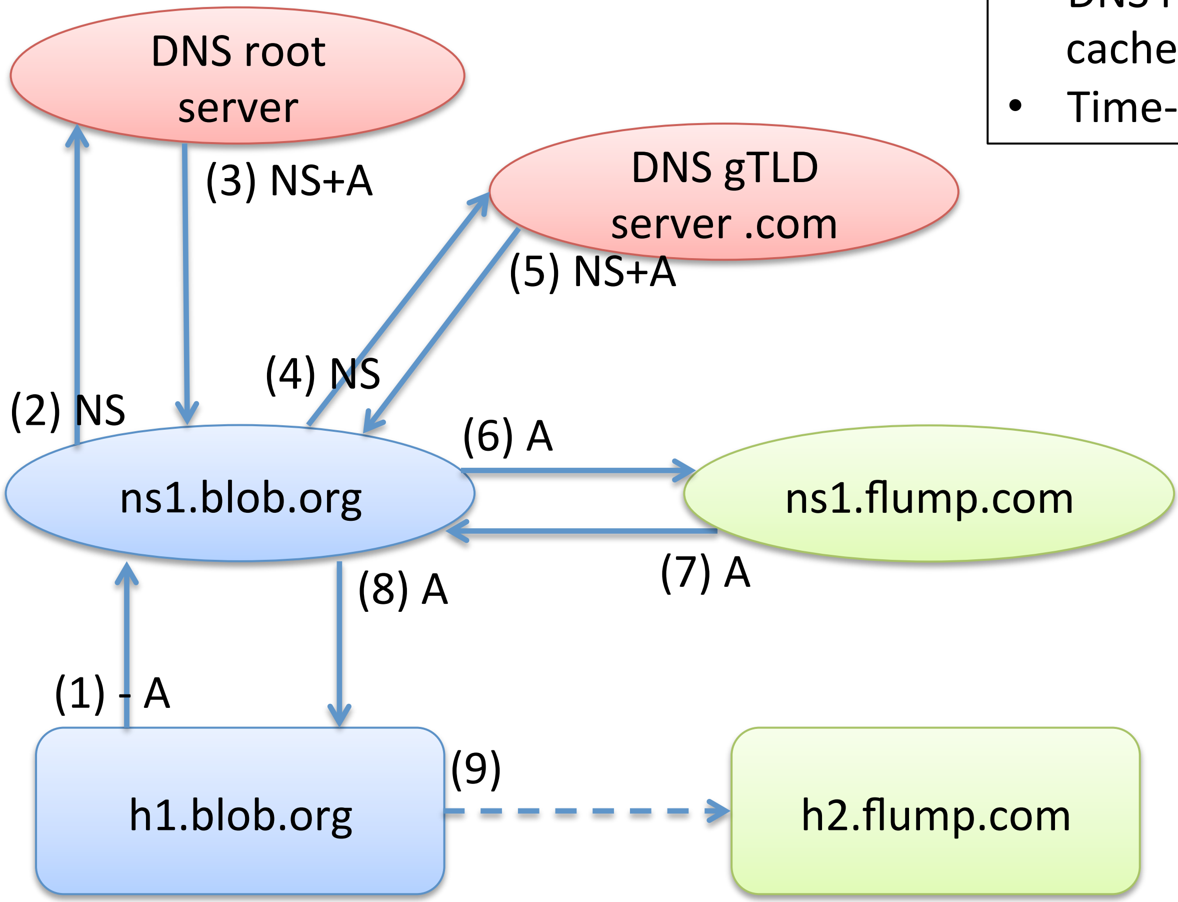
- ILNP – http://ilnp.cs.st-andrews.ac.uk/
- ILNP nodes update (topological) Locator value in DNS as they move to different IP networks.
- Movement could happen at any time, so cached Locator values could become stale.
- We need to reduce caching of such values, ideally to zero, so there are no stale values.
- **Is zero caching practical?**
  Test with A records – nearest equivalent for ILNP.

DNS – a summary [1]

- Scaling – architecture mechanisms:
  - Hierarchical name-space
  - Administrative zones across the name space
  - Delegation of lookups across zones
- Scaling – engineering mechanisms:
  - Administrative **zones** organised around naming hierarchy
  - DNS protocol permits **redirection** (**referral**) to another **name server** for resolving a name
  - Returned results are **cached**
- **Administratively organised, spatio-temporal caching hierarchy**

DNS – a summary [2]

h1.blob.org to lookup h2.flump.com



- DNS resource records (RRs) can be cached for different times.
- Time-To-Live (TTL) value for each RR.

NS RRs (and associated A RRs) returned at steps (3), (5) and (7) can have different TTLs to the A RR returned at step (8).

DNS data

App. data

DNS caching recommendation

- RFC1034 (STD13) on TTL values, p13:

```
… the realities of Internet performance
suggest that these times should be on the
order of days for the typical host.
```

- Today, TTLs generally have large values, e.g.: www.sjtu.edu.cn uses TTL of 43200s (12h)
- Small TTL values (i.e. a few seconds or lower) considered “bad” (some exceptions …)

(Non-)Effectiveness of DNS caching

University of St Andrews

- Jung, J., Sit, E., Balakrishnan, H., and Morris, R. 2002. *DNS performance and the effectiveness of caching*. IEEE/ACM Trans. on Networking. Vol. 10, No. 5 (Oct. 2002), pp. 589-603.
- DNS caching has reduced effectiveness 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**

1. The DNS zero TTL challenge
2. **Experiment configuration**
3. Observations and analyses
4. Round-up

DNS experiments at StA [1]

University of St Andrews

- Experiments in Q4/2009
- Modify TTL values of records in operational DNS server at School of CS, St Andrews
  - 4 DNS servers: Windows ActiveDirectory
  - ~500 DNS clients: Windows, Linux, MacOSX, BSD
- TTL values for successive 7-day periods during normal semester:
  - changed DNS TTL on ActiveDirectory
  - TTL values used: 1800s, 30s, 0s
- Configured clients not to cache.

DNS experiments at StA [2]

University of St Andrews

- 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 data-sets: overall meta-data

Data set name	TTL [s]	Duration [s][1]	Total DNS packets captured[2]	Number of A record requests for 67 servers[3]
dns1800	1800	601,200	41,868,522	2,004,133 (4.8%)
dns0030	30	601,200	71,105,247	2,648,796 (3.7%)
dns0000	0	601,200	55,868,573	4,501,590 (8.1%)

[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, retransmissions etc

[3] servers that were active during the 3 weeks of data capture

2009 data-sets: internal meta-data

Data set name	TTL [s]	Duration [s][1]	Total DNS packets captured[2]	Number of A record requests for 67 servers[3]
dns1800-i	1800	601,200	29,486,362	792,339 (2.7%)
dns0030-i	30	601,200	54,097,231	951,485 (1.8%)
dns0000-i	0	601,200	30,555,305	1,419,782 (4.7%)

[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

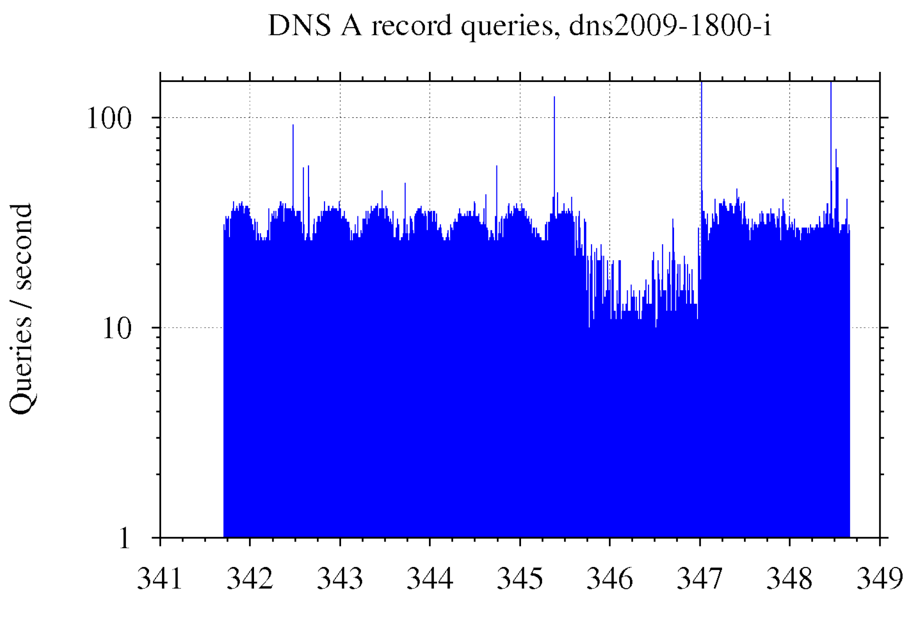
[2] includes all request and response packets to/from port 53 (TCP and UDP), including erroneous requests, retransmissions etc

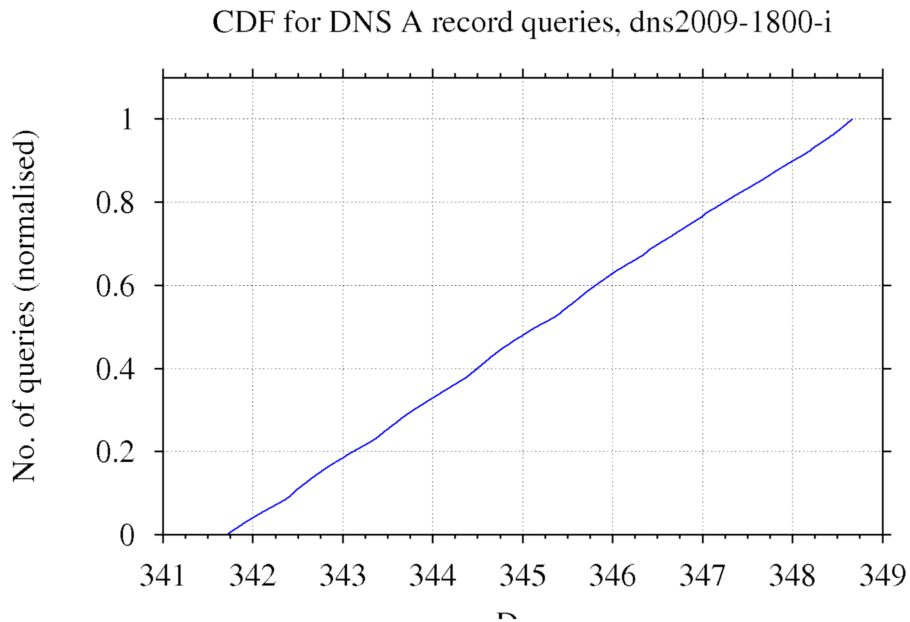
[3] servers that were active during the 3 weeks of data capture

1. The DNS zero TTL challenge
2. Experiment configuration
3. **Observations and analyses**
4. Round-up

dns1800-i: A queries TTL=1800s

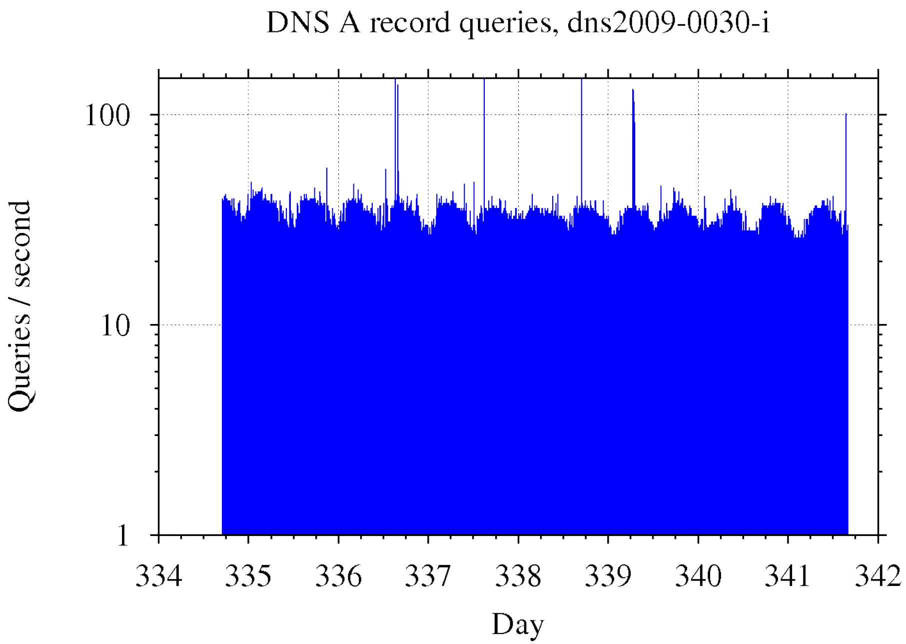
University of St Andrews

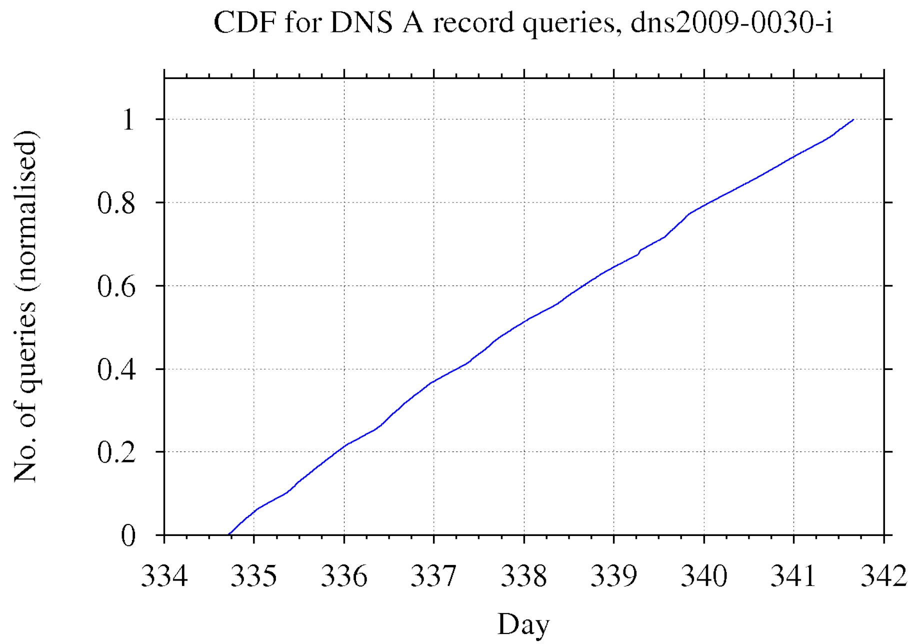




Mean: 1.31 request/s
Std Dev: 2.98 requests/s
Max: 176 requests/s

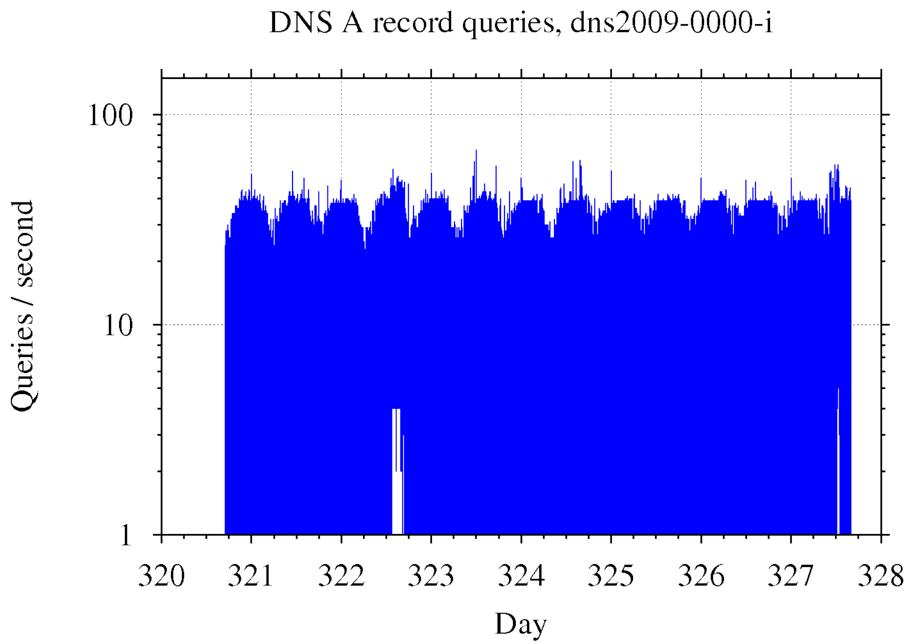
dns0030-i: A queries TTL=30s

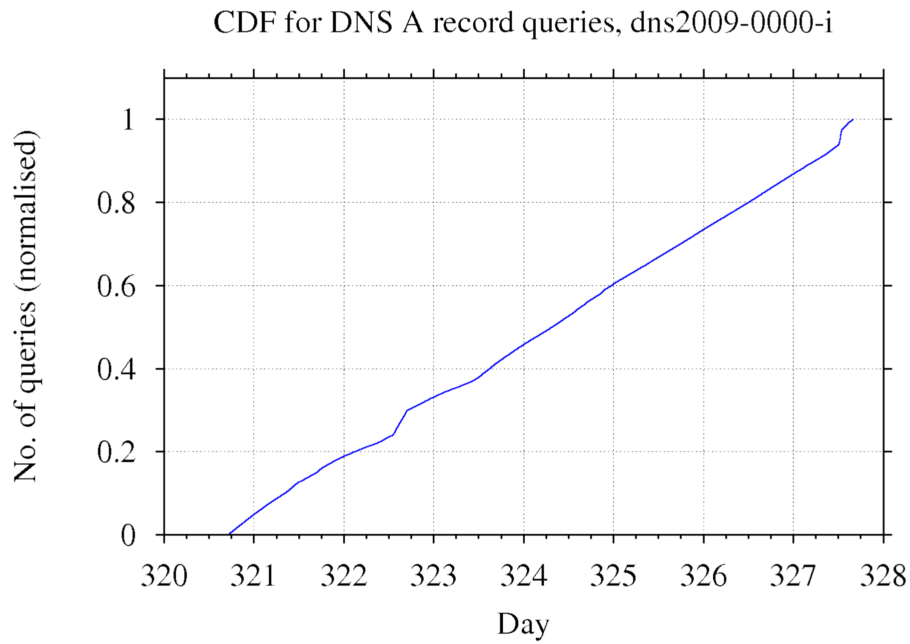




Mean: 1.58 request/s
Std Dev: 3.57 requests/s
Max: 168 requests/s

dns0000-i: A queries TTL=0s





Mean: 2.36 request/s
Std Dev: 3.48 requests/s
Max: 68 requests/s

2009 Summary of basic statistics

University of St Andrews

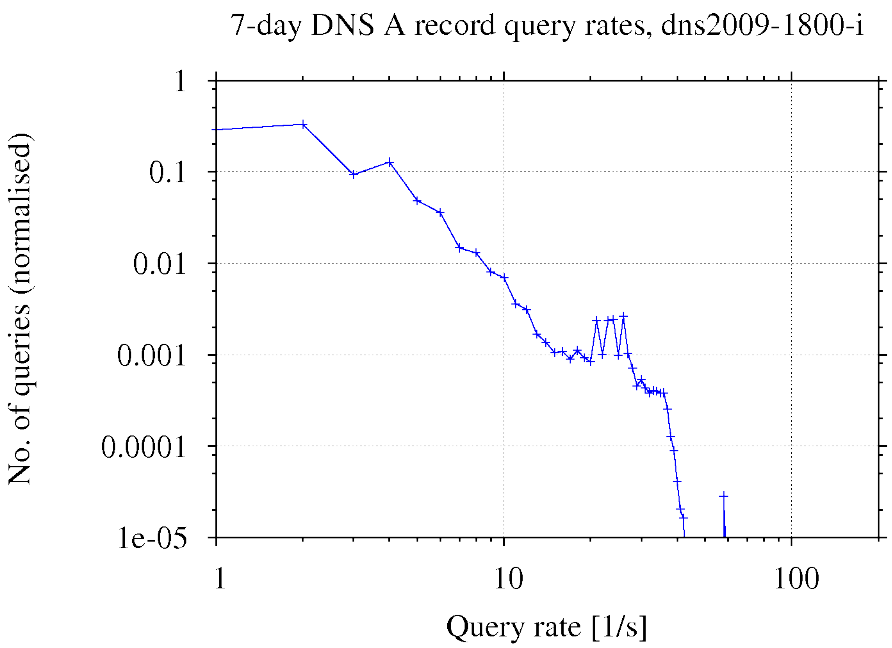
Data set name	Mean [reqs/s]	Std Dev [reqs/s]	Maximum [reqs/s]
dns1800-i	1.31	2.98	176
dns0030-i	1.58	3.57	168
dns0000-i	2.36	3.48	68

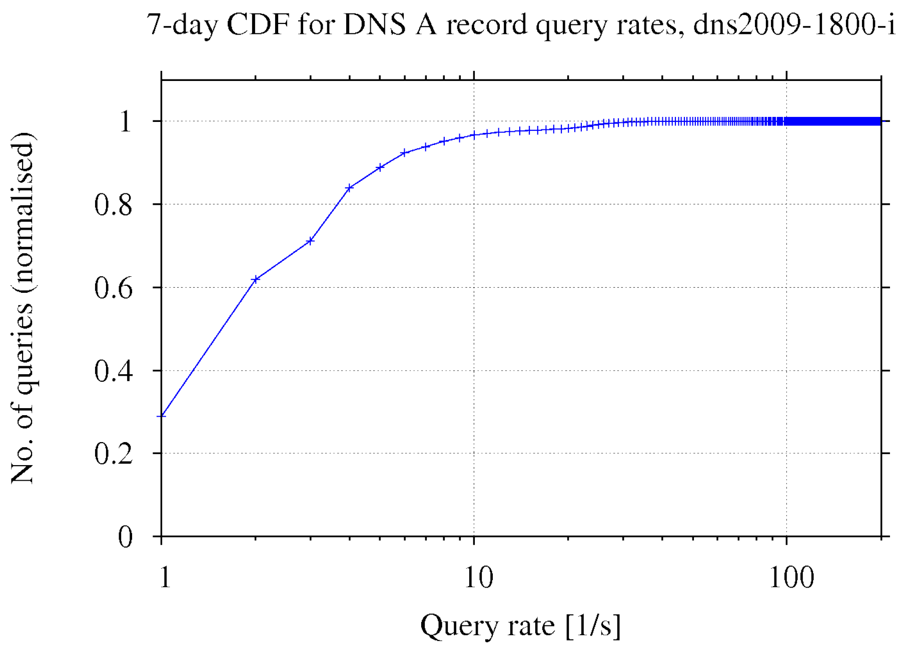
60x drop in TTL values results in
⅓x increase in A record requests.
0 TTL gives ~**2x increase.**

2009 Basic spectral analysis

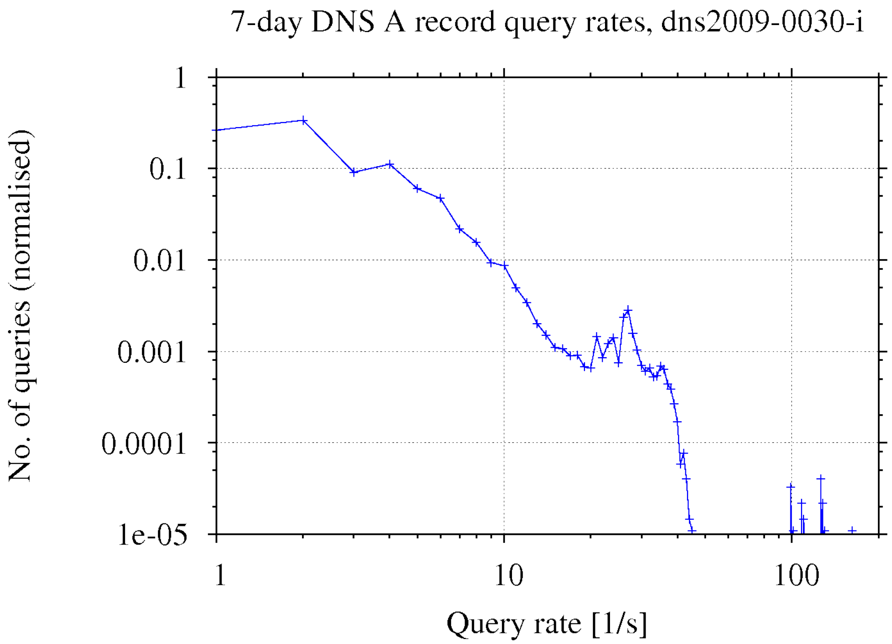
- Create approximate periodogram by counting occurrences of bucket sizes:
  - have used 1s bucket
  - so size of bucket, *n*, is number of requests/s
  - count occurrence of buckets of size *n*
- Comparison of periodogram:
  - shows changing dynamics of request rates
  - gives a better view of the trends in request rates

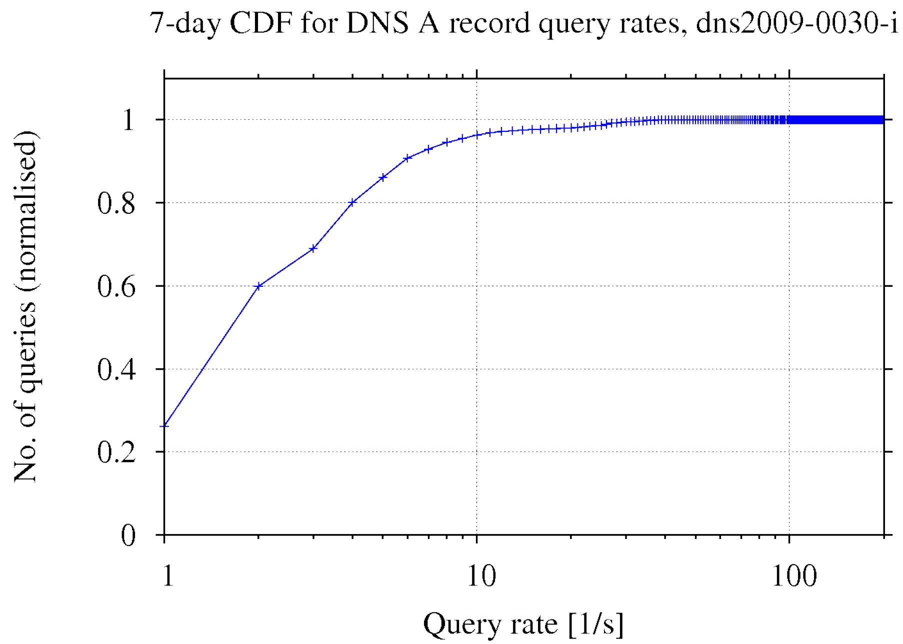
2009 periodograms: 1800s



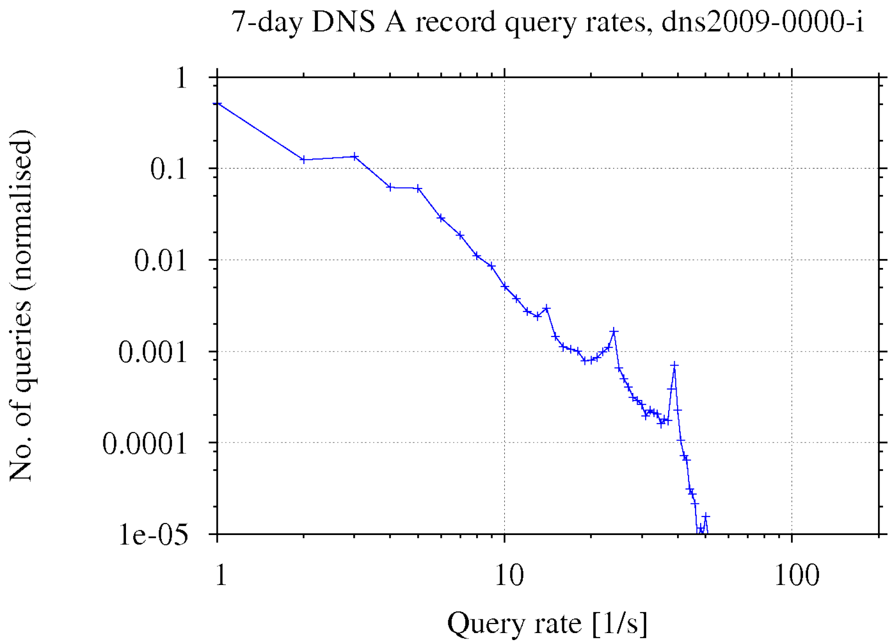


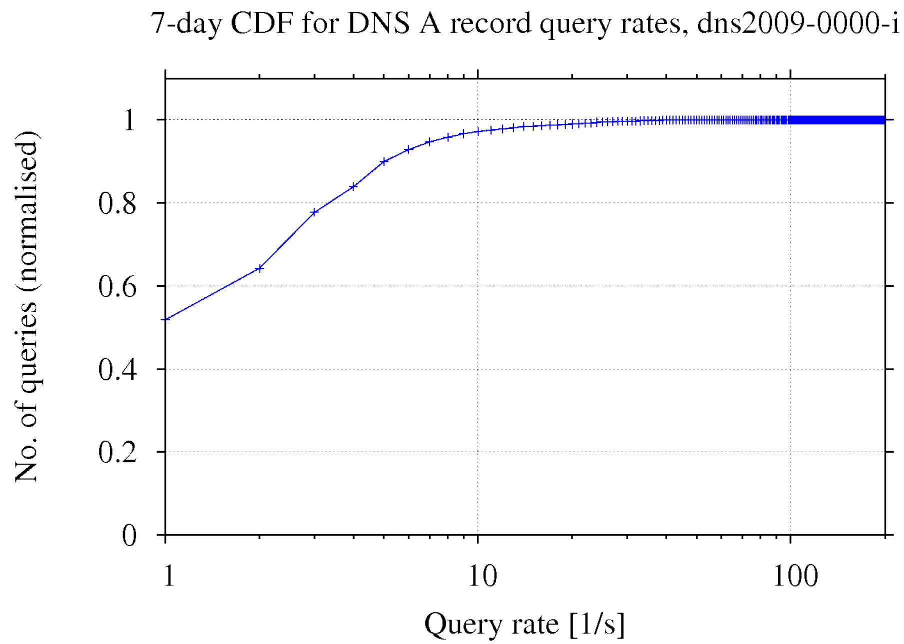
2009 periodograms: 30s





2009 periodograms: 0s





2009 Summary of basic statistics

University of St Andrews

Data set name	Mean [reqs/s]	~95% [reqs/s]	~99% [reqs/s]
dns1800-i	1.31	8	22
dns0030-i	1.58	8	24
dns0000-i	2.36	8	15

~95% centile is the same and is of a low value (8 reqs/s)

University of St Andrews

1. The DNS zero TTL challenge
2. Experiment configuration
3. Observations and analyses
4. **Round-up**

Who would set DNS TTLs so low?

University of St Andrews

- Real A record values for some services:
  - TTL = 60 seconds: yahoo
  - TTL = 20 seconds: akamai
  - **TTL = 0 seconds: St Andrews, Computer Science**
- 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.

Future work

- More in-depth analyses of traces:
  - possibly some controlled experiments
- Repeat experiments at other sites
- For mobility:
  - Secure DNS Dynamic Update
  - DNSsec (authenticated responses)
- Have started some discussions with various industrial collaborators.

Summary and Conclusion

University of St Andrews

- Summary:
  - Zero TTL values for edge-site DNS A records possible
  - DNS load with zero DNS TTLs seems manageable
  - (Indeed, 1s TTL is good, perhaps better than zero)
- Conclusion:
  - DNS A records with very low TTL seems practical

Acknowledgements

- Thanks to:
  - Stuart Cheshire (Apple)
  - Dave Thaler (Microsoft)
    for information on OS-specific features of DNS operation in end-hosts
- Attendees at NANOG50 and IETF79
- Comments from GI2011 reviewers
- **A Very Big Thanks to:**
  - **the Systems Admin Group at cs.st-andrews.ac.uk for implementing DNS TTL changes**