University
of
St Andrews

DNS Caching: Running on Zero

Saleem Bhatti
School of Computer Science
University of St Andrews

Using DNS for networking

- (From ILNP – a work in progress.)
- Use of names and name resolution for:
  - Mobility (host and network)
  - Multi-homing and fail-over
  - TE options
  - Load balancing
  - VM management/migration
  - Others …
- Enhanced use of DNS compared to today:
  - Dynamic updates of DNS records
  - Cache times (TTL) of records may need to be reduced

(Non-)Effectiveness of DNS caching

- 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**

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
  - TTL values used: 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

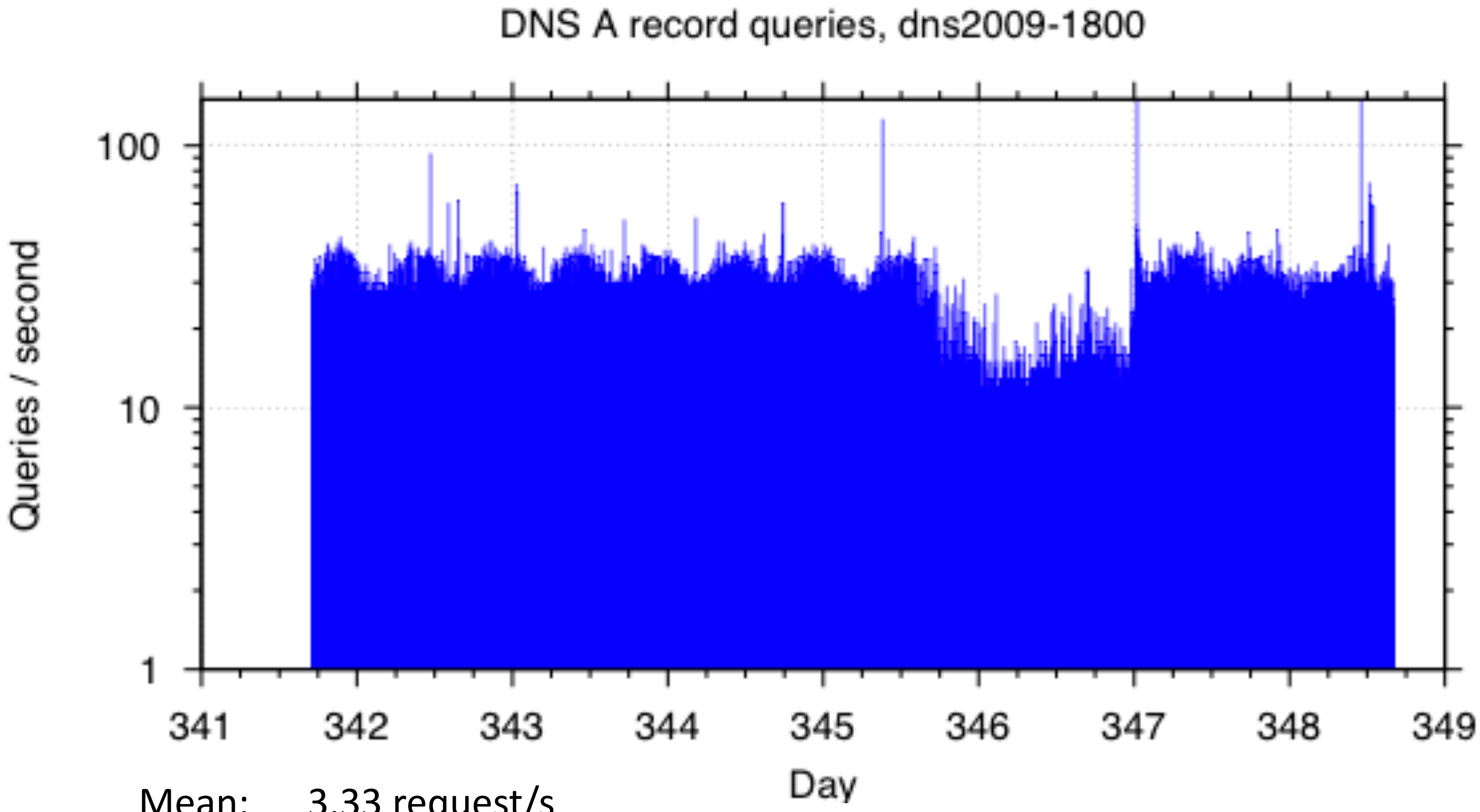
Data set name	TTL [s]	Duration [s][1]	Total DNS packets captured[2]	Number of A record requests for 67 servers[3]
dns1800	1800	601,200	41,868,522	2,004,133
dns30	30	601,200	71,105,247	2,648,796
dn15	15	601,200	56,472,027	3,240,675
dns0	0	601,200	55,868,573	4,501,590

[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

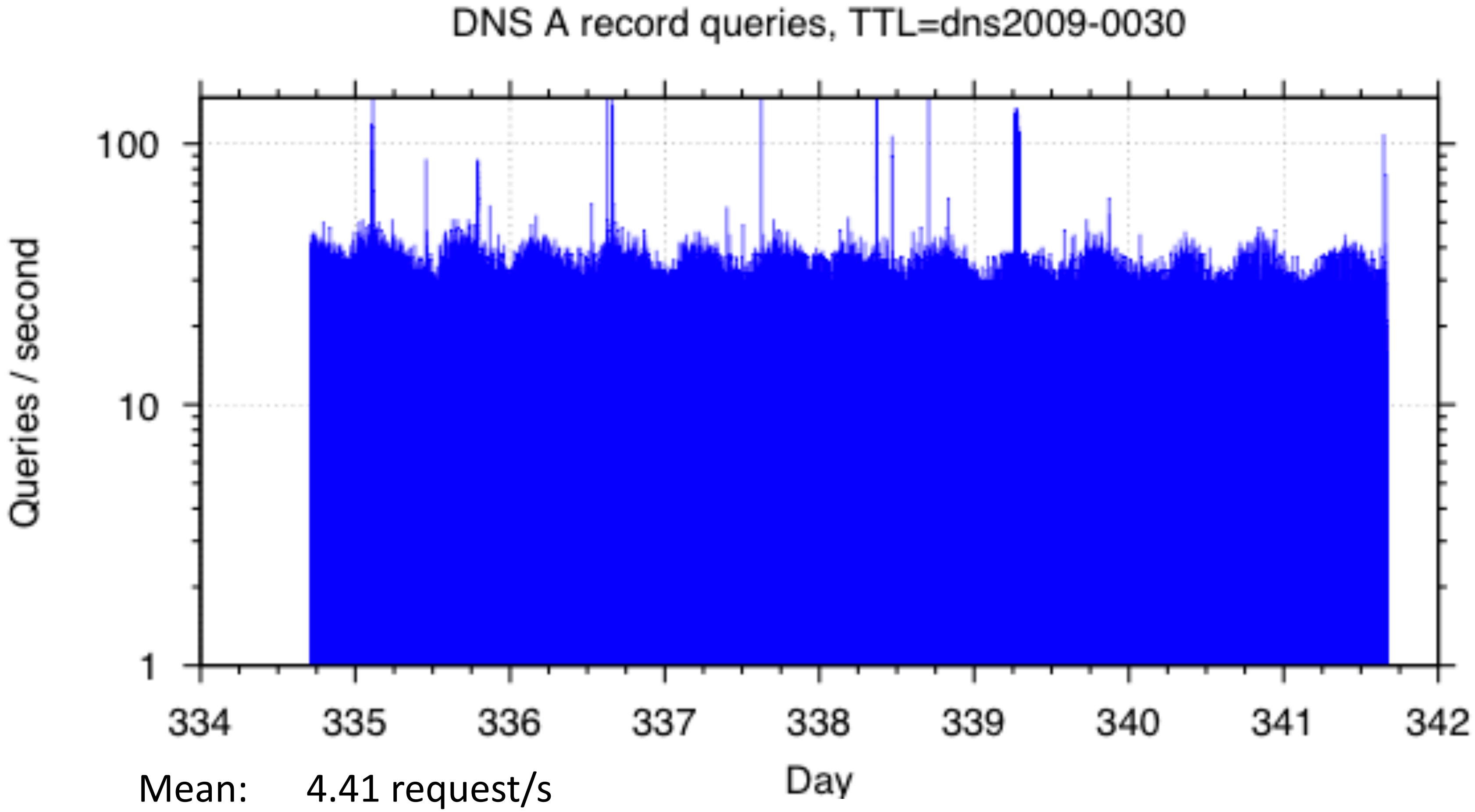
[3] servers that were active during the 4 weeks of data capture

dns1800: A record requests TTL=1800s



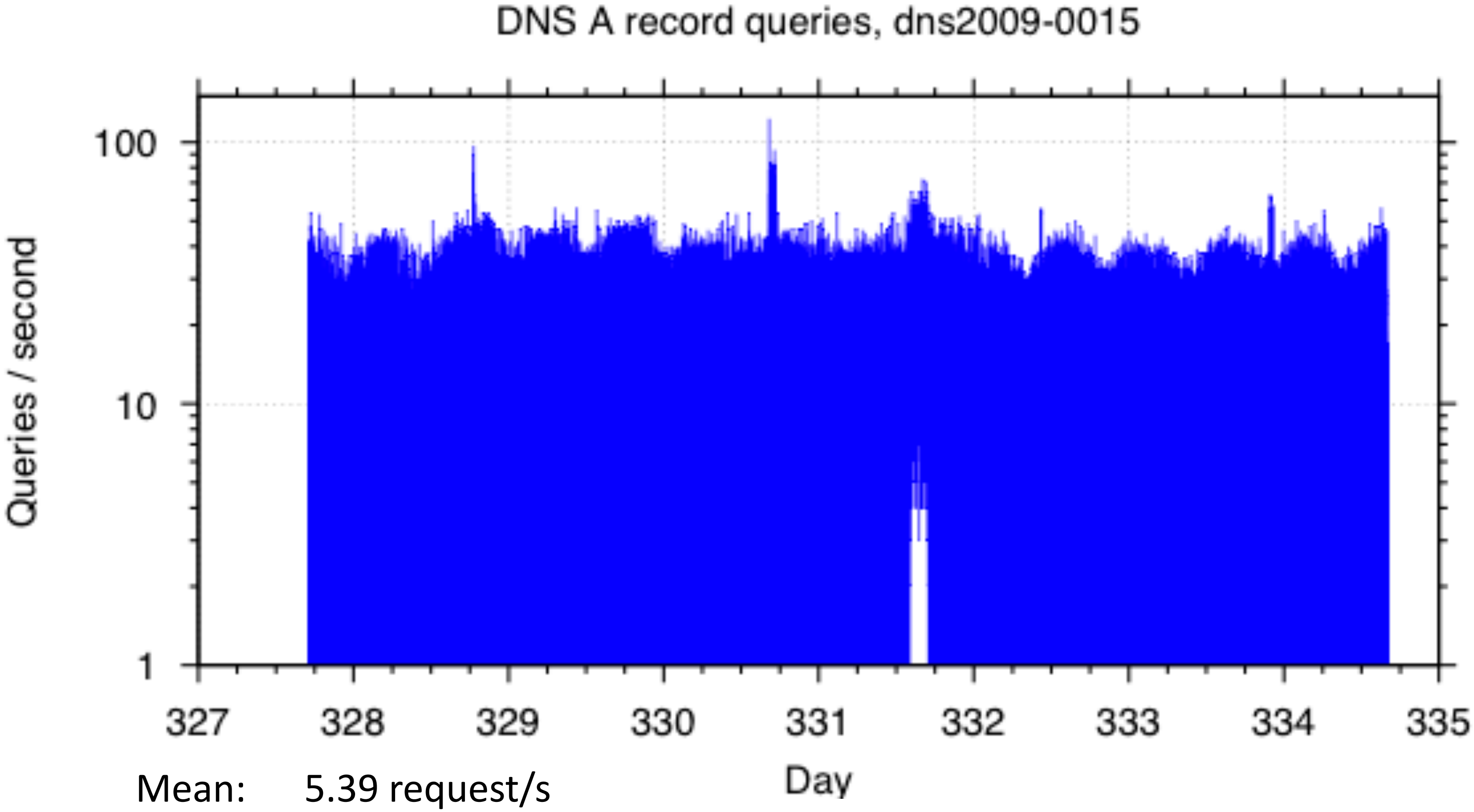
Mean: 3.33 request/s
Std Dev: 3.47 requests/s
Max: 183 requests/s

dns30: A record requests TTL=30s



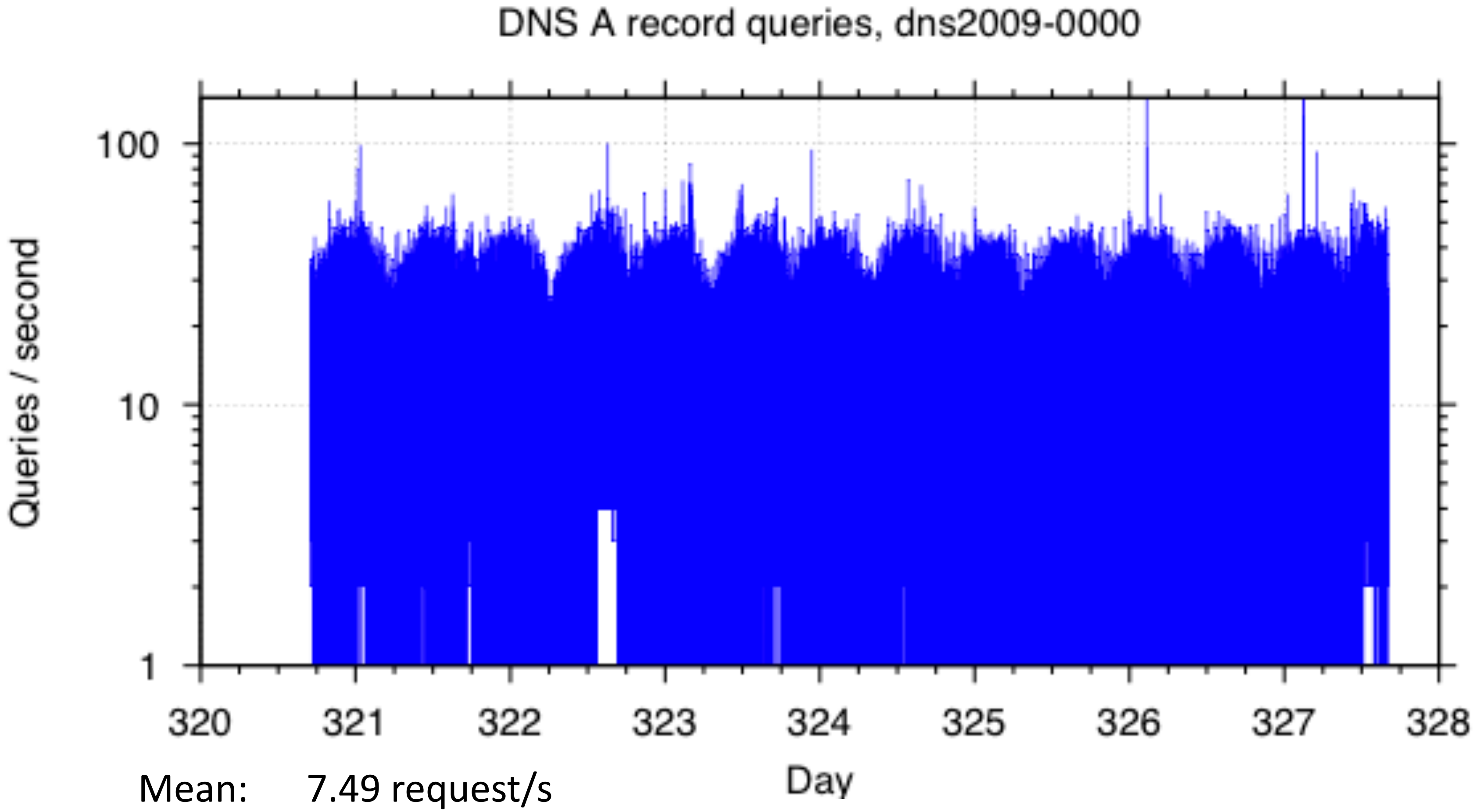
Mean:	4.41 request/s
Std Dev:	4.27 requests/s
Max:	261 requests/s

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



Mean: 7.49 request/s
Std Dev: 4.93 requests/s
Max: 3.69 requests/s

2009 Summary of basic statistics

Data set name	Mean [reqs/s]	Median [reqs/s]	Std Dev [reqs/s]	Maximum [reqs/s]
dns1800	3.33	3	3.47	183
dns30	4.41	4	4.27	261
dns15	5.39	4	4.85	123
dns0	7.49	7	4.93	369

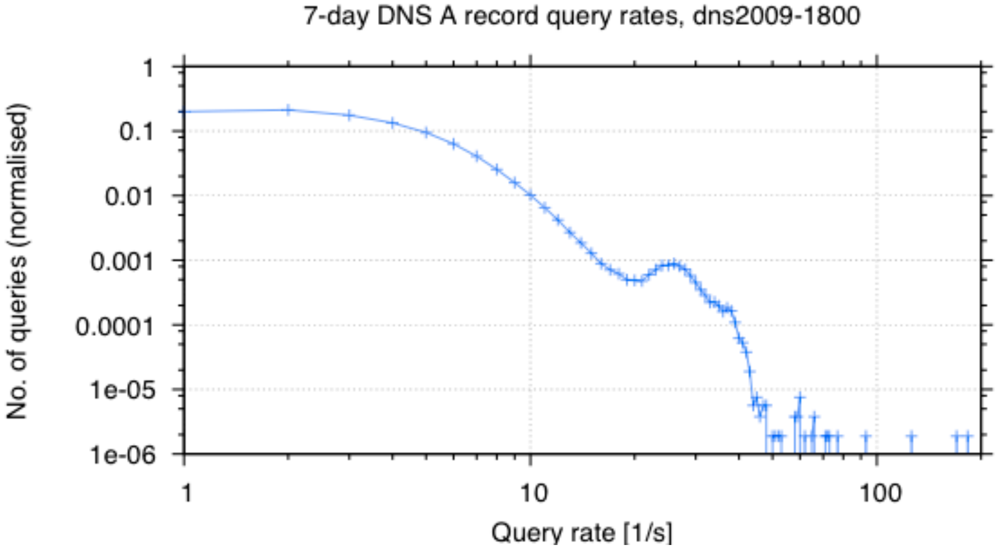
60x drop in TTL values results in **⅓x increase** in A record requests.

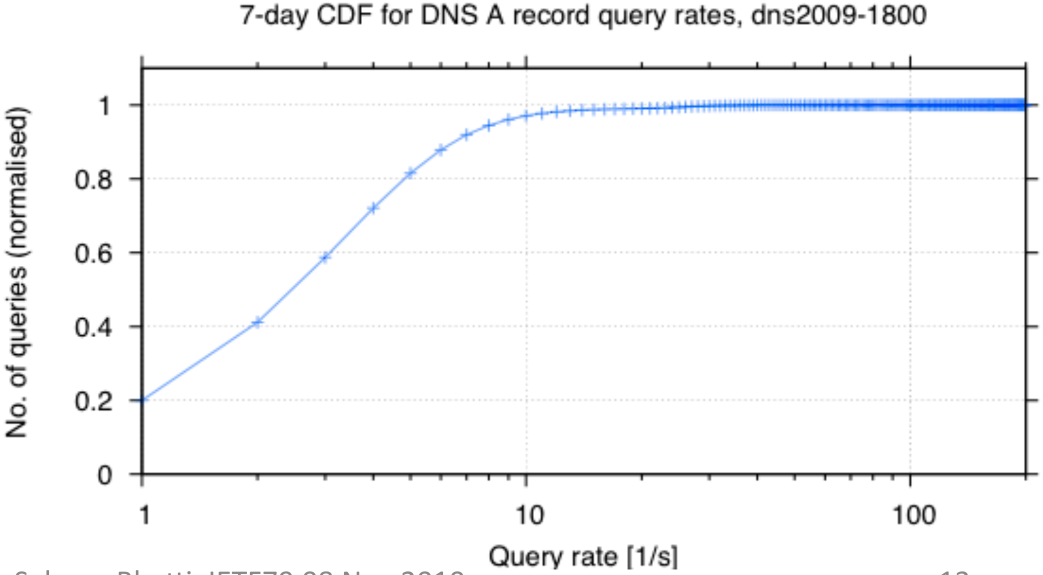
0 TTL gives (only) **2¼x increase.**

2009 Basic spectral analysis

- Create approximate periodogram by counting occurrences 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 view of the trends in request rates

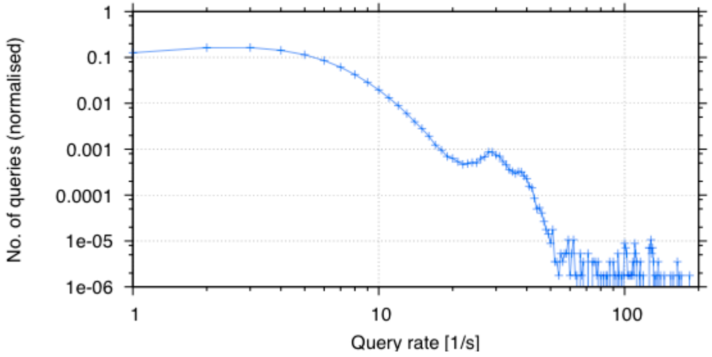
2009 periodograms: 1800s …



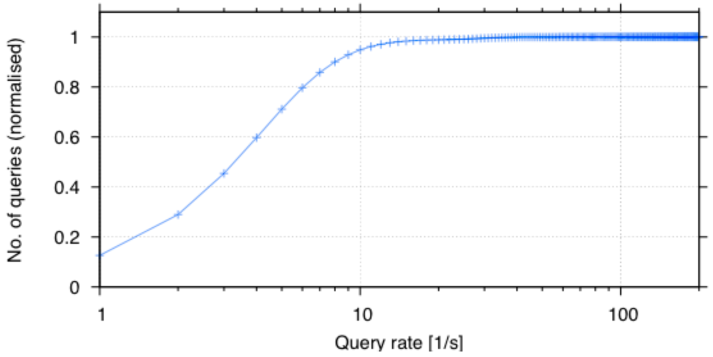


… 30s, 15s, 0s

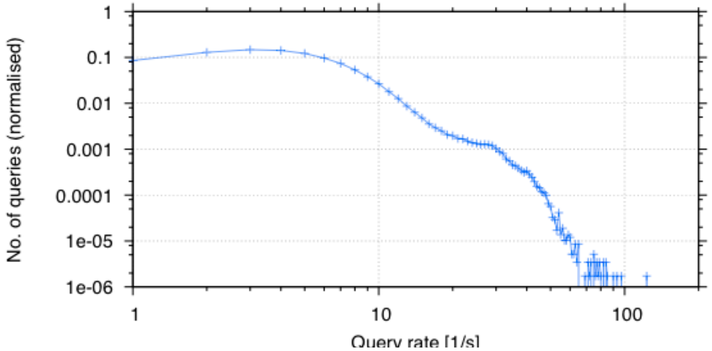
7-day DNS A record query rates, dns2009-0030



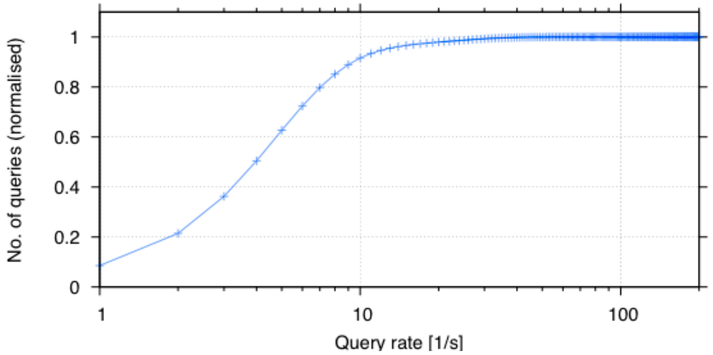
7-day CDF for DNS A record query rates, dns2009-0030



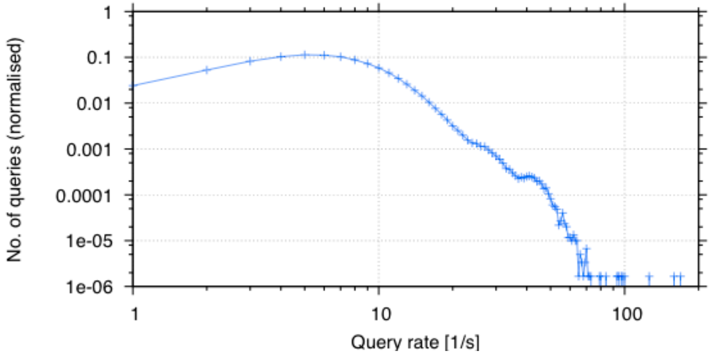
7-day DNS A record query rates, dns2009-0015



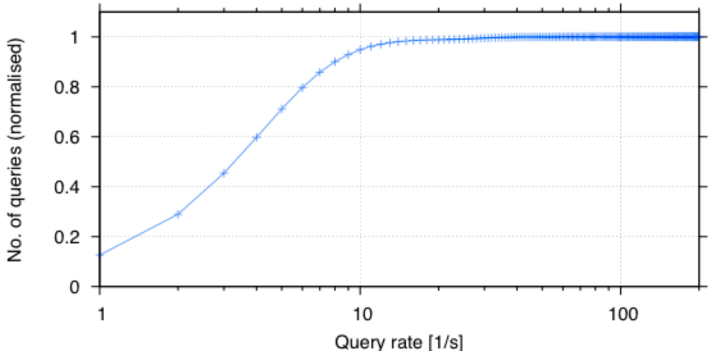
7-day CDF for DNS A record query rates, dns2009-0015



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



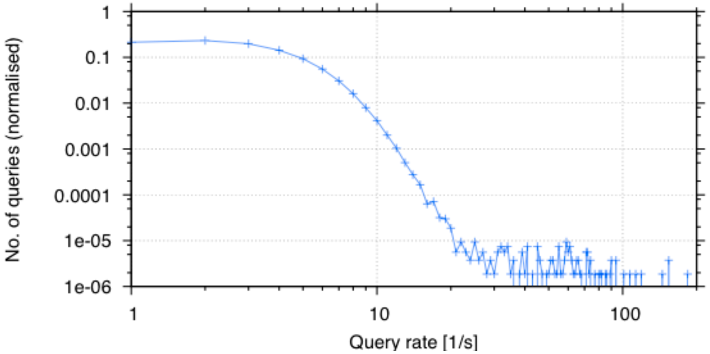
7-day CDF for DNS A record query rates, dns2009-0030



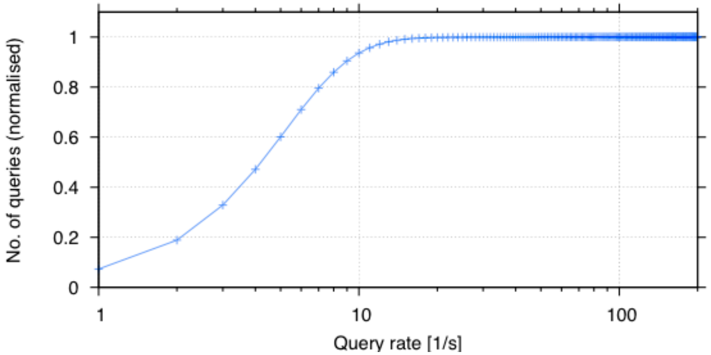
External – 30s, 15s, 0s

University of St Andrews

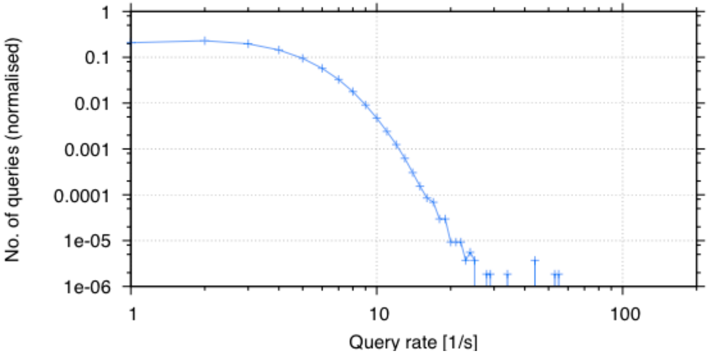
7-day DNS A record query rates, dns2009-0030-o



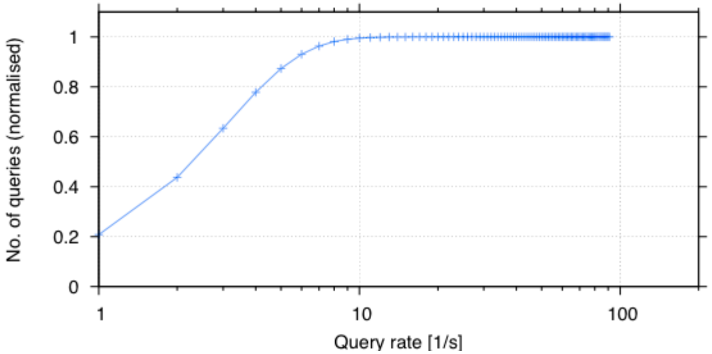
7-day CDF for DNS A record query rates, dns2009-0000-o



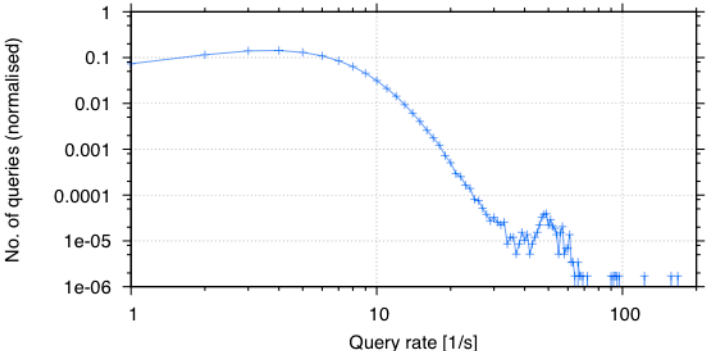
7-day DNS A record query rates, dns2009-0015-o



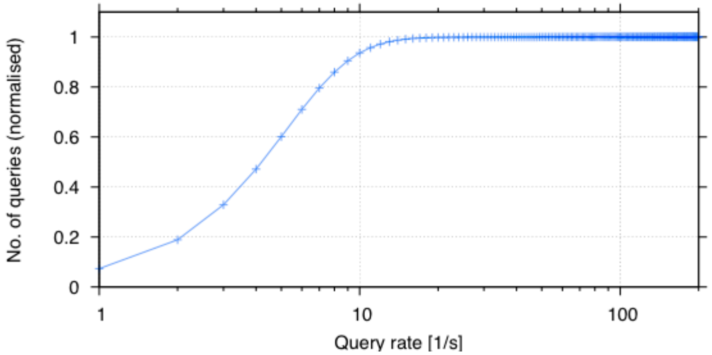
7-day CDF for DNS A record query rates, dns2009-0015-o



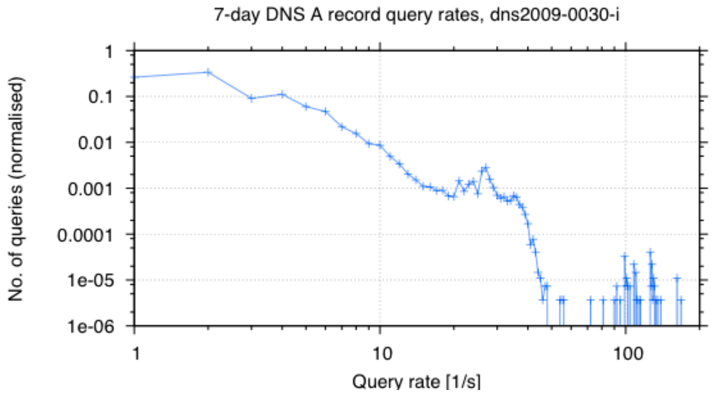
7-day DNS A record query rates, dns2009-0000-o

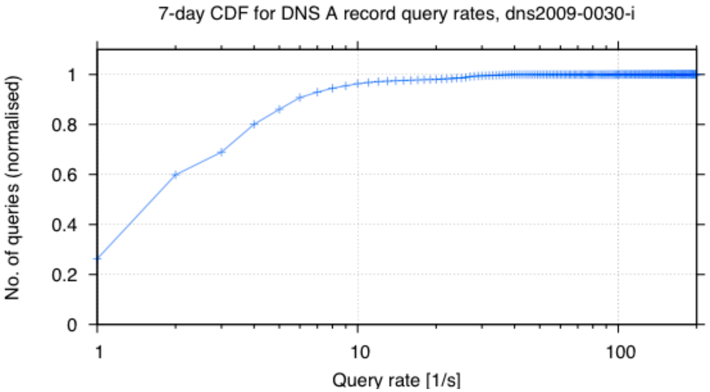


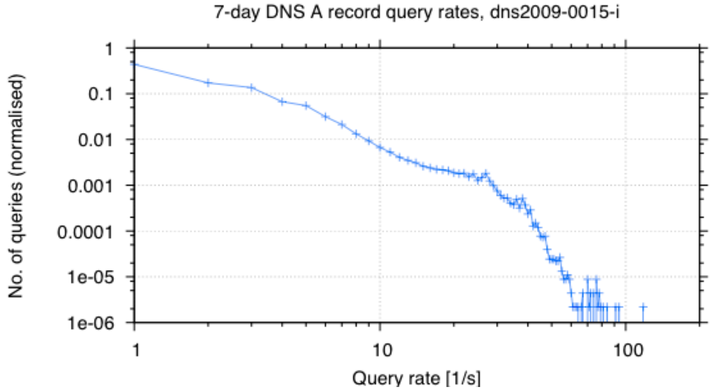
7-day CDF for DNS A record query rates, dns2009-0000-o

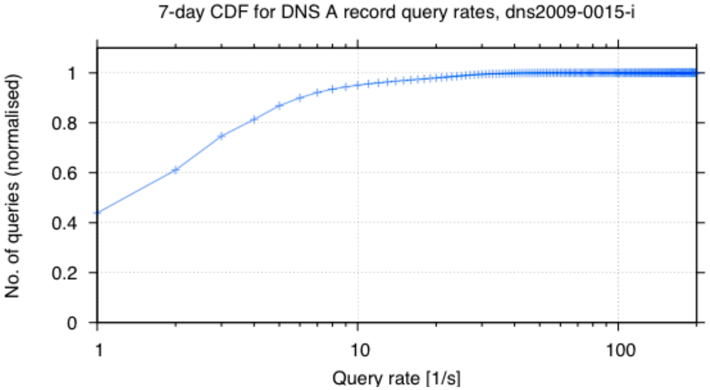


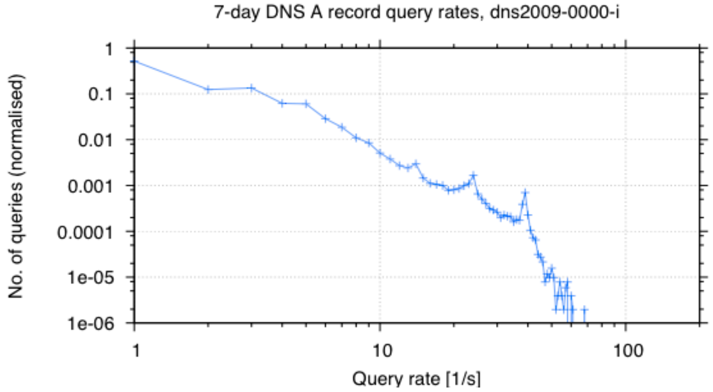
Internal – 30s, 15s, 0s

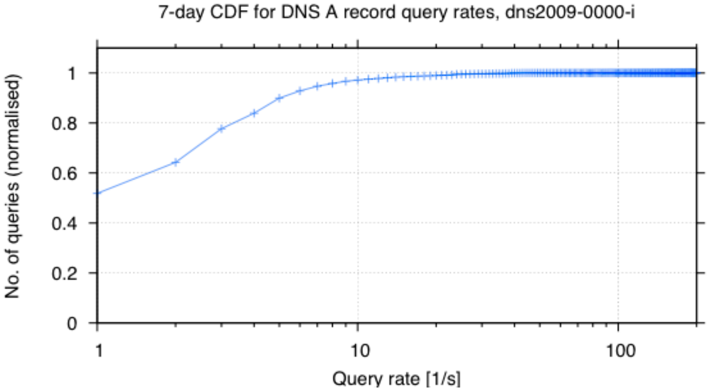

7-day DNS A record query rates, dns2009-0030-i


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


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


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


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


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

Who would set DNS TTLs so low?

- 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.

Acknowledgements

- Thanks to:
  - Stuart Cheshire (Apple)
  - Dave Thaler (Microsoft)

  for information on OS-specific features of DNS operation in end-hosts
- **A Very Big Thanks to:**
  - **the Systems Admin Group at cs.st-andrews.ac.uk for implementing DNS TTL changes**

Summary and Conclusion

University of St Andrews

- Summary:
  - Zero TTL values for edge-site DNS records possible
  - DNS load with zero DNS TTLs seems manageable
  - (Indeed, 1s is good, perhaps better than zero …)
- Conclusion:
  - Frequent DNS access for records with very low TTL seems practical
- Future work:
  - Scale experiment: analyses of larger DNS deployments
  - Impact of the use of Secure DNS Dynamic Update and cryptographic authentication of DNS look-ups