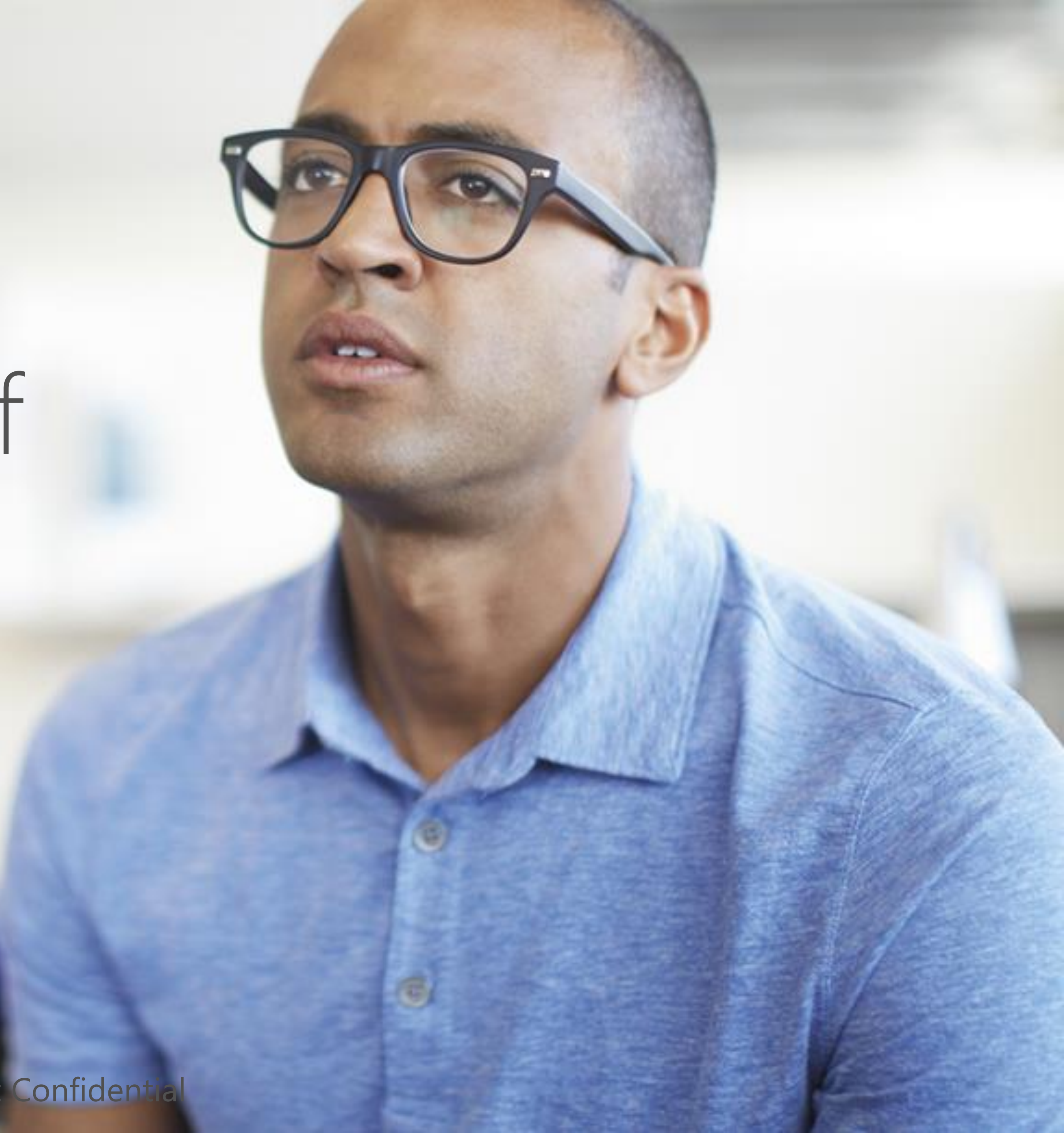
Negative Caching of DNS records

Stephan Lagerholm
IT Service Engineering Manager, Microsoft Azure

Joe Roselli
Senior IT Service Engineer, Microsoft Azure



Agenda

Background, Problem statement and RFC – Stephan Lagerholm

Experiments – Joe Roselli

Conclusion and Discussion - Everybody

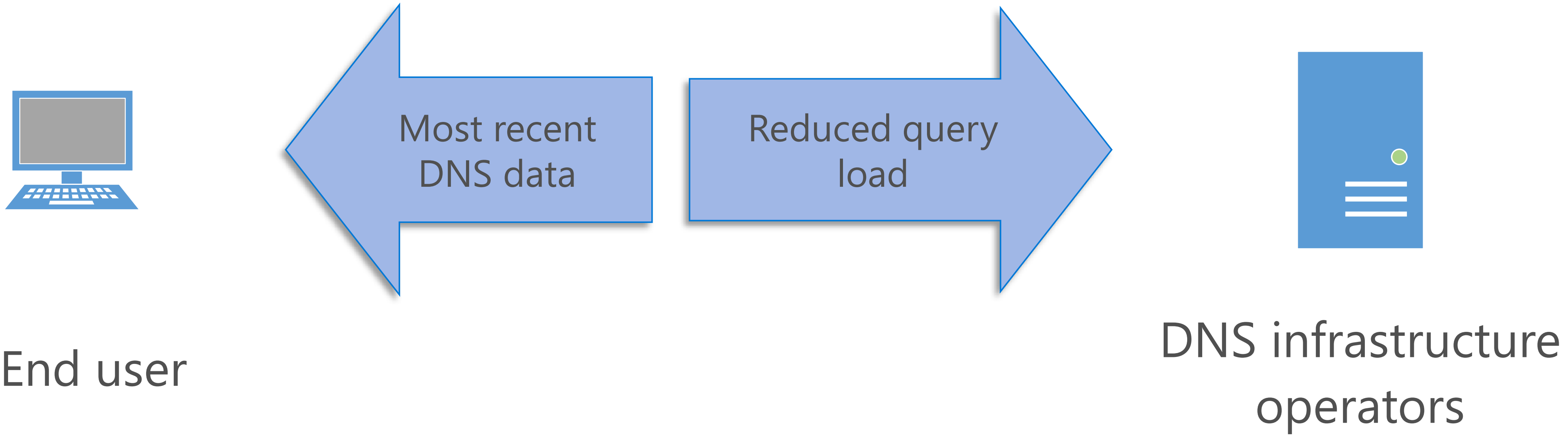
Background

- We realized that we need to better understand for how long negatively cached records can be found on the Internet.
- The problem discussed here is similar to Duane Vessels OARC Phoenix presentation in 2013 “An Open Resolver view of the New York Times Very Bad Day”
- We are discussing negative caching **NOT** “normal” or SERVFAIL caching.
- Examining negative caching, we are examining both the Authoritative and Recursive DNS behaviors.

Questions to Answer

- How long will the lack of a DNS record be cached on the Internet?
- Is negative caching an efficient method of improving the DNS experience?
- What additional experiments and documentation are needed to better understand the negative caching process?

Negative caching competing interests



Most recent DNS data

Reduced query load

End user

DNS infrastructure operators

A look at SOA records

```
msft.net.               86400 IN SOA ns1.msft.net. msnhst.microsoft.com. (
                                2015013001 ; serial
                                7200       ; refresh (2 hours)
                                900        ; retry (15 minutes)
                                2419200    ; expire (4 weeks)
                                3600       ; minimum (1 hour)
                                )
```

- SOA records are included in queries resulting in no response for both NXDOMAIN **and** EMPTY NOERROR
- NXDOMAIN and EMPTY NOERROR (NODATA) TTLs are referenced as **Negative TTLs** for the remainder of this presentation

RFC 2308

Section 4 - SOA Minimum Field

Despite being the original defined meaning, the first of these, the minimum TTL value of all RRs in a zone, has never in practice been used and is hereby deprecated.

…being the TTL to be used for **negative responses**, is the new defined meaning of the **SOA minimum field**.

Section 3 – Negative Answers from Authoritative Servers

The TTL of this [SOA] record is set from the **minimum of the MINIMUM field of the SOA record and the TTL of the SOA itself**, and indicates how long a resolver may cache the negative answer.

Section 5 – Caching Negative Answers

As there is no record in the answer section to which this TTL can be applied, the TTL must be carried by another method. This is done by including the SOA record from the zone in the authority section of the reply. When the authoritative server creates this record its TTL is taken from the minimum of the SOA.MINIMUM field and SOA's TTL. **This TTL decrements in a similar manner to a normal cached answer**…

Recursive Server behavior

Default Negative TTL

Brand	Default max Negative TTL
Windows	15 minutes
Bind	3 hours
Unbound	1 day
Power DNS recursor	1 hour

RFC 2308: "Values of one to three hours have been found to work well and would make a sensible default."

Testing 3 popular recursive DNS servers

Microsoft Windows DNS:

```
dig asdsadas.google.com    +noadd +nocomments +noquestion
google.com.          300    IN     SOA     ns1.google.com. dns-admin.google.com. 86056494 7200 1800 1209600 300
```

Bind:

```
dig asdsadas.google.com +noadd +nocomments +noquestion
google.com.          60     IN     SOA     ns1.google.com. dns-admin.google.com. 86056494 7200 1800 1209600 300
```

Unbound:

```
dig asdsadas.google.com +noadd +nocomments +noquestion
google.com.          600    IN     SOA     ns1.google.com. dns-admin.google.com. 86056494 7200 1800 1209600 300
```

Experiments

Authoritative check methodology

- Query for Alexa top 1,000,000's name servers to check for both RFC compliance and TTL duration
- Query all Authoritative Servers for domain SOA and dummy record
- Compare Negative TTL response to SOA TTL and Minimum TTL values
- Results are based on Name Servers for compliance check:

  Name Server goes on the non-compliant list if it gives an unexpected Negative TTL value.

Example Results and Categories

Domain	NameServer	SOA TTL	SOA MinTTL	Negative TTL	Compliance Status
vanderbilt.edu	ip-srv1.vanderbilt.edu	86400	3600	3600	Compliant
melissaaustralia.com.au	ns2.bdm.microsoftonline.com	3600	3600	1	NotCompliantLow
clevermarket.gr	ns2.lighthouse.gr	86400	7200	86400	NotCompliantHigh
nashville.gov	ns1.nashville.gov	3600	3600	3600	Matching

TTL Check Results

Negative TTL length (s)	% of responsive top domains
TTL <= 900s	27%
900 < TTL <= 3600	28%
3600 < TTL <= 86400	44%
TTL > 1 day	1%

For 45% of the top domains, one accidental record deletion will have some impact on the Internet for more than an hour, sometimes more than a day.

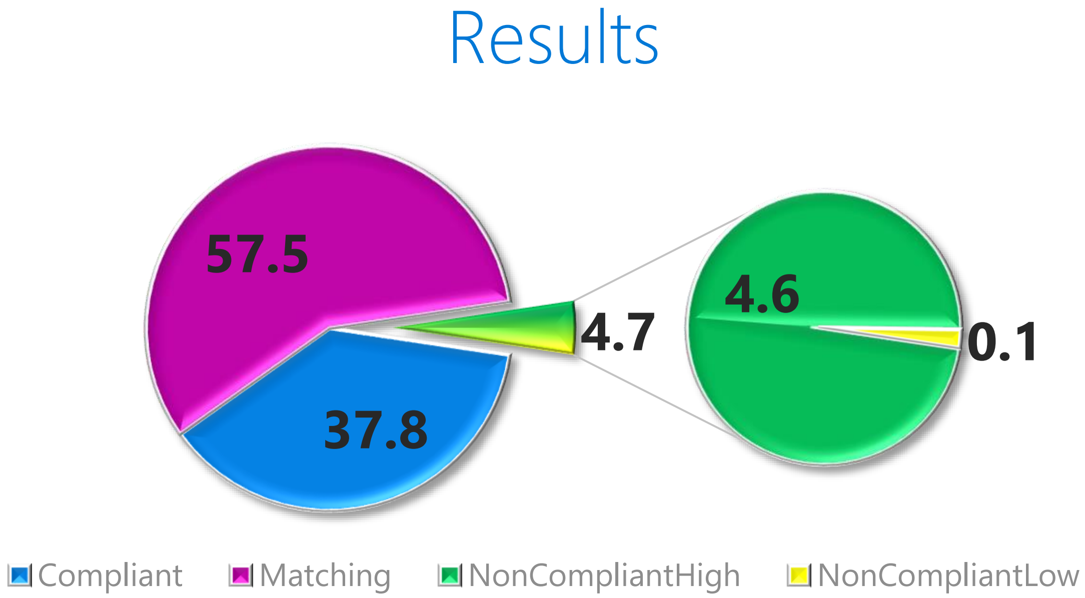
Authoritative Check Results

Alexa top million

High Negative TTL	4.6%
Low Negative TTL	0.1%
SUM not RFC compliant	**4.70%**
Confirmed RFC compliant	37.80%
Matching	57.50%

TLDs

High Negative TTL	0.7%
Low Negative TTL	0%
SUM not RFC compliant	**0.7%**
Confirmed RFC compliant	49.6%
Matching	49.7%



Authoritative experiment conclusions

- A large percentage of domains are set for high Negative TTL values.
- Over 18,000 name servers are responding with an unexpected Negative TTL value.
- The combination of high and unexpected Negative TTL values have the potential for slowing full Internet-wide recovery times for mistakenly removed records.
- Recommendation for Authoritative DNS server operators: Set the SOA TTL and MinTTL values to the *same value* (preferably < 3600 when there's capacity).

Negative Cache experiments

Production authoritative observation

- Response rate investigation on an active authoritative zone.
- Compare results of different Negative TTL settings on a zone where all records have a 3600s TTL to find extra error volume if any.

Negative TTL	Total Number of queries over 2 hours	Number of queries resulting in an error response over 2 hours	% of queries resulting in an error response
900s	102,631	66,038	64%
3600s	69,915	29,056	42%
28800s	49,978	7,148	14%

Negative Cache experiments

Recursive server observations

Disable negative caching on	Increase in Queries
Validating Resolver	20%
Non-Validating	8.5%

- Clear benefits for enabling negative caching from a load perspective on a recursive resolver.

ORNS Experiment methodology

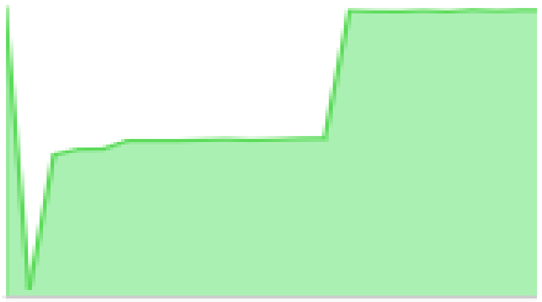
- Using over 2,500 known ORNS through a monitoring tool – 15 minute minimum time between checks.
- Take a known good record with known Negative TTL values and verify all ORNS resolve the record as expected.
- Remove the record on the authoritative servers and verify that all ORNS give a negative response.
- Restore the record and time how long it take each ORNS to give a valid response.

ORNS results

Negative TTL	Time to 90% ORNS recovery	% taking full TTL to recover	Time elapsed from record restoration to full ORNS recovery
900s (15 minutes)	15 minutes	0.4 %	20 – 35 minutes
3600s (1 hour)	50 minutes	0.9%	65 - 80 minutes
28800s (8 hours)	2 hours 45 minutes	0.3%	9 hours 30 minutes
86400s (1 day)	2 hours 45 minutes	0.1%	1 day 30 minutes

ORNS Results Explanation

Phase	Time after simulated outage	% of resolvers that recovered	Comment
1	15 min	50%	Resolvers cache records for a maximum of around 15 minutes.
2	3 hours	98%	We suspect that resolvers that recover after 3 hours are default configured Bind servers. Bind has a default setting for max-ncache-ttl of 3 hours.
3	1 day	100%	The remaining 2% of resolvers recovered after 1 day.



Microsoft Confidential

JR

Conclusions

- Full Internet recovery time for a missing record can be slower than expected due to Authoritative and Resolver behaviors.
- Surprises can be minimized on the Authoritative-side if the SOA TTL and MinTTL are set to the same values.
- About 5% of Authoritative servers are not RFC 2308 compliant in how they handle the Negative TTL. Recursive software may consider being “noble” and check that the authoritative server handled RFC 2308 properly.
- RFC 2308 was written in 1998 and *a lot* of things have changed since then. We have moved to a more interactive Internet.
- Call to further discuss the best balance for reducing downtime without significantly increasing query load.

JR

Microsoft