For DNS server caching, what is the ideal TTL?

Many factors affect how to set time to live (TTL) for DNS servers. Learn more, plus how BlueCat Edge's TTL features can bolster your network.

As a DNS administrator, do you have default time-to-live (TTL) settings for your DNS servers?

In an informal poll of BlueCat customers, answers included eight hours, two hours, one hour, and 15 minutes. Some admins vary their TTLs depending on whether the DNS traffic is on their internal network or going out to the internet.

Is there a most common TTL? An ideal? Should it be long or short? What *is* the magic answer?

(If James Bond had a say in the matter, he'd insist that there's no time to die—only time to live.)

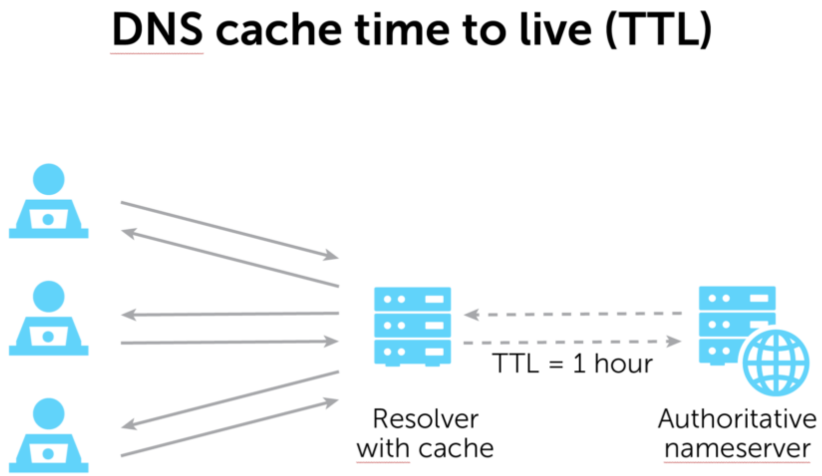
This post will touch on what time to live is in the context of DNS caching. Then, it will discuss the factors to consider when deciding on the right TTL for your network. Finally, it will explore how the TTL-related features of BlueCat Edge can make your network more secure and resilient.

The small group of IT professionals who discussed these ideas is part of BlueCat's open DDI and DNS expert conversations. All are welcome to join Network VIP on Slack.

What is TTL for DNS caching?

When discussing TTL in the context of DNS caching, the basic idea is that cached responses increase efficiency. DNS record information can be cached locally inside a device's stub resolver or somewhere in the DNS server infrastructure. Cached information circumvents further steps and more quickly delivers responses to DNS queries.

Once the TTL on a cached record expires, a recursive DNS resolver must begin the lookup process anew. It will have to resolve the DNS query via an authoritative nameserver.



Separate from DNS caching, TTL is also used to ensure IP packets have a limited lifetime on a network. (For IPv6 addresses, it is called the hop limit.) This information is contained in the header field of the packet. It specifies the maximum number of landings on network devices (known as hops) that the packet can make en route to its destination. These TTLs and hop limits are measured in seconds.

When discarding a packet with a TTL or hop limit of one or zero, a router may send an internet control message protocol (ICMP) error message to the source.

Deciding on the right time to live for your network

If one thing is certain, it's that the TTL requirements for every network will vary.

"It's one of those areas in DNS where it's almost like there isn't a correct answer," says Pablo Garrido, Senior Technical Product Manager for BlueCat Edge. "It's a use-case correct answer."

Making it short—but not too short

Short TTLs have their advantages. They can increase DNS propagation speeds, help update systems more quickly, and make load balancing more effective.

However, customers agreed that permanently setting a TTL that's too low is a recipe for failure. A very short TTL can tax your downstream servers more than necessary. Depending on the situation, the response for a given query may not change for hours. Constantly pinging for the latest answer is inefficient.

Fellow IT community member Frank Denis also recently laid out his argument for avoiding TTLs that are too low.

"You pay a penalty. And your consumers for that service pay a not-insignificant penalty in the overall lifespan of the service delivery model for that," one customer said. "If you're a solution designer—cloud or enterprise, it doesn't really matter to me—and you design something that somehow requires a near-zero TTL, you flunked. You flunked your job of doing a proper design for this solution."

On the other hand, a TTL that is too long means that you may be serving obsolete responses.

How customers set their minimum TTLs

One customer sets their lowest TTL based on the average minimum length of time it takes to migrate a service from one server to another. Another sets a minimum TTL and, if users ask for something below that, considers requests on a case-by-case basis. They may lower it for certain events to support rapid service change operations. But they set it back to the required minimum level once the event concludes.

"If you have an actual DNS architecture, you understand where the caching is occurring and how your TTLs will interact and interplay," the customer said. "Then you can make intelligent decisions as to what your default TTL should be or where the minimums should be."

Improve security and resiliency with BlueCat Edge TTL features

Speaking of no time to die, the namespace feature on BlueCat Edge bucks that trend and allows queries to have a second life. If a DNS query receives an NXDOMAIN response the first time, it can try a second set of servers downstream. Called Intelligent Forwarding, it gives the query two lives at the very least. (Because, hey, you only live twice, right?)

Edge features also give users more security and resilience when it comes to DNS TTL in particular.

Enhanced security with single-click cache flushing and short TTL

Edge service points act as the 'first hop' for client devices–essentially a caching forwarder. But BlueCat has found that customers who want to create policies to block certain network traffic have been beholden to the TTLs of the domains for those queries. You can enact a policy to block them, yes. But it can only take effect once the queries in that cache have expired.

With Edge, you can flush the cache on a service point with a single click. This allows a new policy to take effect immediately. It's important to note, however, that this feature is not for every use case. It is reactive and heavy-handed, as it does clear the entire cache.

As a more proactive alternative, Edge also features a namespace setting called short TTL. By just checking a box, all of the DNS requests that travel through that namespace automatically get a 60-second TTL. Edge overrides the TTL in the response and makes it 60 seconds. As a result, users get the most relevant responses to their queries.

It's important to note that you can set up domain lists to control what FQDNs will be affected by these rules. So you can, for example, exclude internal domains that you don't want to be controlled by a shortened TTL.

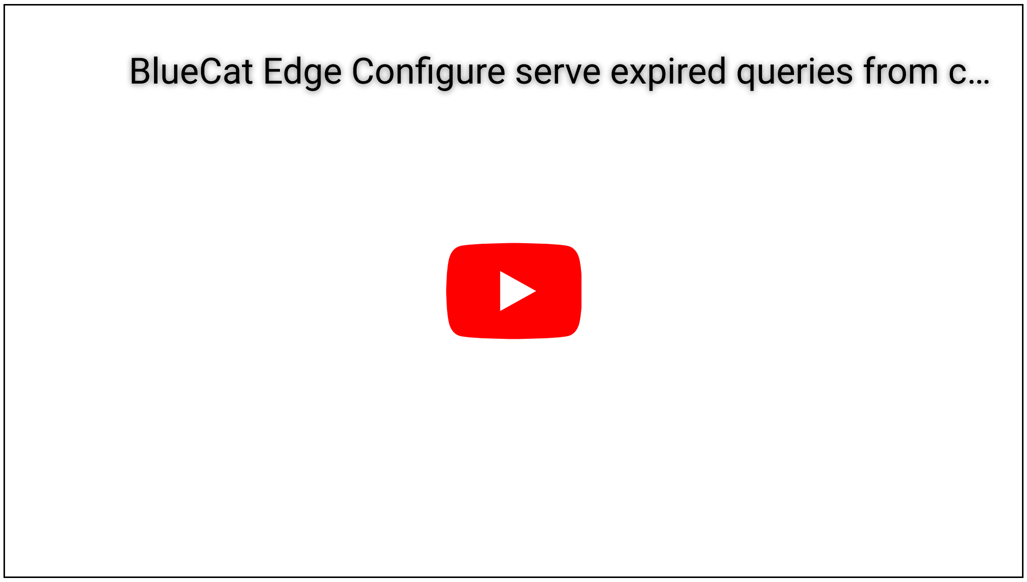
More resiliency by serving expired queries from cache

To enhance resiliency, Edge can serve queries from cache after a response's TTL has expired. It's as simple as selecting a radio button.

The serve expired feature doesn't affect TTL per se. But it extends the life of an expired response in the cache until a new response is obtained. Extensions can last for at least one hour, up to 24 hours.

In cases of branch installations, serving expired queries from cache for a full day after expiring can help ensure a seamless experience. Or, consider if the forwarders for an internal namespace are non-responsive. By extending life of the expired response, Edge can still answer and make it appear to the client as if that outage isn't present.

Here's more on how to configure server-expired queries from cache at the Edge service point:



BlueCat Edge Configure serve expired queries from c…

The potential to make TTL values fully configurable

For future releases of Edge, BlueCat is considering making TTL values fully configurable.

For a short TTL, is 60 seconds too long? Maybe it's 15 seconds that you want. Or, when serving expired queries from a cache, maybe one hour is too long and 15 or 30 minutes would be more effective.

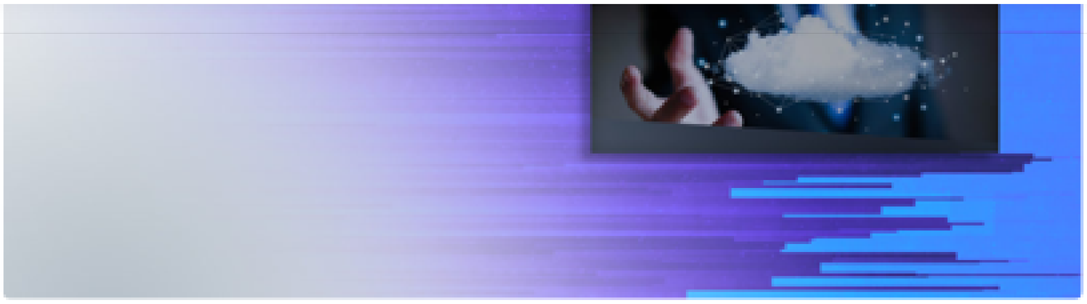
Share your thoughts on this potential cache control feature in Edge, or how you configure TTL in general, in Network VIP.

About BlueCat

Keep system issues at bay with health checks

Two levels of BlueCat support offer health checks to analyze customers' system data for potential problems and fix them before they take networks down.

Read more



Cloud Networking Dysfunction

Why do seven in 10 enterprises struggle to realize the full value of the cloud? This EMA research contains clues and guidance on how to change that.

Learn more

We're using cookies on this site to improve your experience. Cookies help us learn how you interact with our website, and remember you when you come back so we can tailor it to your interests.

You can find out more about cookies and usage on our privacy policy page.

Accept

Decline

Why BlueCat?

Adaptive DNS

Products and services

Core network services

Network automation

Network security

Hybrid cloud management

Services

About

Partner with BlueCat

Our leadership team

Our board of directors

Careers

Channel partners

Technology partners

BlueCat for government

Resources

Case studies

Strategic content

Technical content

Glossary

News and press

Blog

Support

Communities

Contact

Privacy License agreements Cookie Notice