Edge DNS GSLB

Complement Your Load Balancing and Multi-Cloud Strategy

Contents:

A long time ago, when the internet was far smaller, IP addresses were used to reach services. The introduction of qualified names for the services has simplified usage, particularly when it comes to browsers. DNS has been created to manage the hierarchy of names and fully qualified domain names which appear on the web to name all the services. We still use today the base principles defined at the early stage of the Internet.

Destination

DNS is used quite often, in the background, as an easy solution to get the technical information required for reaching an application or a service. DNS gives the destination information, in this case an IP address, potentially a port number and for some more specific usages it can provide rich information like signature or authority proof. Once the IP address is known, a TCP or UDP session can be established in order to transport the data.

Recursion and Caching

For efficiency purposes, the DNS service is hierarchical, distributed and resilient. It uses a smart principle of recursivity in order to ask the server in charge (authoritative server) for the next information required to reach the ultimate goal: getting the IP address of an FQDN. This recursion principle is used on all IP networks - private and as an extension on the whole Internet. The recursion is associated with a caching system which prevents repeated asking of the same information during a period of time considered as small enough for the information to remain valid.

Application Routing

In order to access an application from a client or from another application component, the DNS service is used to convert FQDN into an IP address and optionally gather other technical information. But DNS is not the only component contributing to application access. Of course, the IP network has a massive role to play: it routes IP packets from source to destination and is quite complex. On the path, we can also find security components and more importantly, for applications, some load-balancing solutions. The load-balancers are generally used to distribute the application sessions across multiple servers that can therefore handle load and some failure situations. As part of this overall routing process the DNS plays the major role of «defining the destination», this is why its role is critical and the impacts of the DNS service failing are huge.

Why is a Load Balancer not Enough?

The load balancer is important for application scalability and load sharing because it is on the traffic path near the application servers. It can see in real time if something goes wrong, change the distribution pattern, suppress a server from the pool because it is not responding correctly or fast enough.

The load balancer is located within a datacenter and is the «destination» from the user perspective. It has been designated by the DNS as the destination for the application FQDN. If something bad happens in the datacenter behind the load balancer, it will react accordingly, but if the failure or error is happening between the user and the load balancer, it cannot react. When the IP packet reaches the datacenter, it is too late to send it to another place. And if the frame is lost in between, there is nothing the load-balancer can do.

It is possible to install load balancers on multi-datacenter, multi-cloud architecture, but this is a complex design and topology. It requires specific configuration as well as investment, and bandwidth between datacenters is generally insufficient for all services to work smoothly.

The DNS service can help. Multiple destinations can be associated with a single FQDN and therefore for a single application. If such an application is hosted in multiple datacenters, internal or in the cloud, this round-robin solution existing from the very first days of the DNS implementation can help. The clients, after having asked for resolution of the application name into an IP address (the destination), will randomly choose one of the provided IP addresses and direct their traffic to this selected destination. This will spread the traffic amongst datacenters, the rest of the distribution between servers will be handled by load balancers installed locally in each datacenter.

Why is DNS not Enough?

DNS can distribute traffic with the use of multiple address or service records associated with the same application name. This is very simple to set up, takes advantage of the intrinsic scalability, redundancy and distribution of the DNS service and is directly usable by clients. But load-distribution is not homogeneous. DNS clients and TCP/IP libraries embedded in the various operating systems are not being handled in the same way as these multiple records. Some take just the first one in the proposed list (e.g. Windows), some perform round-robin (e.g. Linux).

With regards to redundancy of the service, the DNS records are not updated automatically whenever a server is added or suppressed from the load-balancing pool. In order to cover this update process, either automation or at minimum some manual operations are required. The caching principle of the recursive DNS service can point clients towards a destination which is no longer able to reply for the service. By lowering the TTL of DNS records, we can limit the impact on services not accessible but still in the cache. In this situation, the application will not be functional for some of the clients and require very complex diagnostic scenarios.

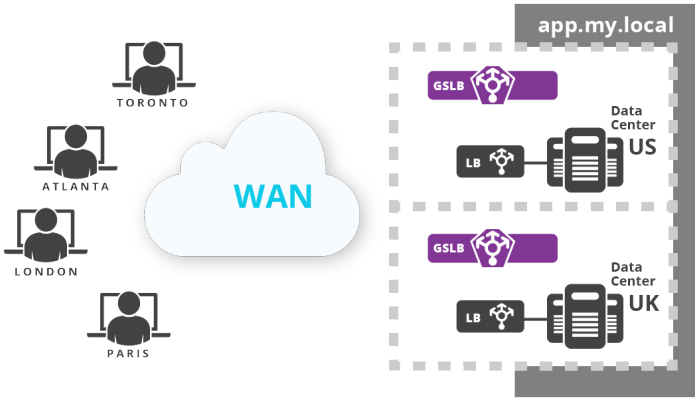
For the DNS to be able to seriously handle failure scenarios, it is necessary to implement health checking and automatic updates, which is not the main purpose of the DNS service.

Last but not least, the DNS is global and gives the same information to all clients. This enables performant caching but is not always appropriate for load-balancing where some may want to provide different answers to different clients, mainly based on their location. It could be interesting for east coast clients to be directed primarily to the datacenter facility located on the east coast, at least to limit long-distance bandwidth usage and to improve latency and consequently the user experience (ping delay is not a problem just for gamers).

In order to help the DNS standard process handle more complex situations like dynamic updates and answers specifically based on location, the DNS GSLB (Global Server Load Balancing) solution has been introduced. It acts as an authoritative DNS server but can dynamically manage the answers from the clients and the other recursive servers based on defined criteria and the health of the destination.

Why is Authoritative DNS GSLB not Enough?

Traditional GSLB servers embedded in a DNS allow answering differently to the same query depending on the location of the client and the availability of the FQDN. The most obvious case, shown in the diagram below, is to direct clients using DNS resolution mechanisms towards different servers hosting the same application or service depending on whether they are in Europe or in North America.



For this purpose of specific and dynamic answering, the DNS GSLB should be authoritative on the DNS zone. This will allow it to change the answer to any query depending on criteria. The drawback of being authoritative is that a service that needs to be answered using the GSLB principle needs to be in this zone. Generally a delegation for a sub-zone is provided to the GSLB server and specific CNAME records are used to move original A and AAAA resolutions from the standard DNS servers to the GSLB. This is not really complex but requires cooperation between teams that are managing the DNS infrastructure and the GSLB one, sometimes they are different. It also requires creating records in advance, managing the TTL and transforming A to CNAME ideally without impacting user traffic.

Centralized GSLB service on an authoritative DNS server also requires specific configuration in order to be able to direct traffic based on the location of the requesting client. If users based in the USA need to be first directed to a datacenter in the USA, it is mandatory to know the IP addresses of all USA clients. On a private network where the IP address plan has been built with considerations other than GSLB, this task requires regular and cumbersome configuration. A misconfiguration of the IP mapping in the GSLB will result in sending the client traffic towards a sub-optimal destination, the service will work but troubleshooting will be complex.

Relying on a centralized service also has an impact on the ability to determine the availability of the server hosting the application. If you want the GSLB to adapt the answers depending on the server health for the benefit of the clients, it requires set up of a health check between the GSLB server and the application servers. Since all these components are generally hosted in datacenters, the ability to know how the client perceives the application answering time - including the network transit from its location to the server - is very challenging. This requires very detailed analysis of the flows between the client and the server, so needs to be located either on the path of all communications or ideally positioned near the client.

Why Edge DNS GSLB is the Smartest Solution

Taking the application traffic routing decision from the edge of the network, where the clients are located, is a very different and smarter approach.

Edge DNS GSLB proposes the best destination address for the requesting client without requiring any complex task to be performed. By being located close to the user, it can share the same viewpoint (or at least a very similar one) and by using the same application protocol for the health check as used by the application service, it is sure to use the identical network path, independent of the use of complex and dynamic systems like load balancers, SD-WAN, hybrid WAN or multi-WAN, Internet VPN or WAN optimization solutions.

Edge DNS GSLB doesn't require dedicated authoritative zone delegation as it can adapt the answers to the client on the fly. Using the health-check results and administrative constraints it is able to automatically determine the best solution for the local clients in order to reach the appropriate server hosting an application.

Having the ability to mix this rich application traffic routing feature with all the features already available in a recursive DNS server brings immediate value for the client. Caching is efficient, information is available locally, filtering can be applied for security, usage of the network for DNS traffic is limited and breakout is respected and even optimized.

The following table shows a comparison between the different solutions, with regards to typical load balancing features required. A position 1, 2, 3 or 4 is assigned for each feature, with 1 being the "best" position.

Subject for Global Load Balancing Purpose	Load Balancer / ADC	Standard DNS multi A records	DNS GSLB authoritative	Edge DNS GSLB
Easy to Set Up	4 very complex to setup globally	1 very easy to do	3 need specific solution (pot. hw/sw)	2 just enable on recursive
Easy to Manage / Administer	4 more complex but more rich	1 easy to administer	3 need zone delegation, change naming or CNAME,	2 easy to add/remove, per record
Easy to Test for App	4 need to move the server behind the LB + IP addressing...	2 need authoritative change	3 need server CNAME/IP + authoritative change	1 rule is taken on the fly for the next request (vs cache)
Ability to Load-balance on Multiple Datacenters	4 not adapted solution	3 easy but no control	2 complex	1 simple configuration
Easy Geolocation of Client	4 in-line, no possible use of the information	3 through views	2 configuration of IP to location mapping to maintain	1 by design
Network Latency Analysis	2 can analyze traffic and RTT in the session	4 not possible	3 not easy, not on the path	1 implicit with health checking from the client side
Network Routing Analysis	3 not easy to know when network is failing	4 not possible	2 not easy (need health check of client - but recursive servers on the path)	1 implicit with health checking from the client side
Server Health Analysis	1 very easy with flow and health check	4 not possible, require external automation	2 can rely on LB information more accurate	3 few health checks method avail, can be added through scripting
Traffic Distribution Between DC	4 need LB on branch sites	3 need management of views, not scalable	2 impacted by recursive cache hierarchy	1 each client can have specific answer
Dynamic Update of Destination Server / DNS Record	1 though session manipulation / transparent proxy	4 need automation and zone/rr manipulation	3 TTL and propagation on the DNS recursion tree	2 decision taken closer to client (ideally 1 hop)

Subject for Global Load Balancing Purpose	Load Balancer / ADC	Standard DNS multi A records	DNS GSLB authoritative	Edge DNS GSLB
Destination Server Failover Transparency	1 can be done without impact	4 no failover	3 need new session for any client behind a recursive (depends on client DNS load-balancing algo)	2 each client may have a specific answer, impact is medium to low if a server switch is required, new session required for some clients
Sticky Session per Client	1 native	4 not possible	3 if intermediate resolver, no information on final client	2 possible per requester client
DNS Delay Impact When Routing to Application	4 high impact if redirecting traffic to another DC	2 easy to add slave, but no control on which will be reached	3 fewer NS for the authoritative zone than traditional dns (cost of the box)	1 near the client
Cumulative Position Score	37 3	39 4	34 2	20 1

Summary of Key Benefits

Edge DNS GSLB routes users to the healthiest and most responsive application server. This brings many benefits, helping to:

- Improve user experience
- Enhance multi-site resiliency
- Simplify disaster recovery plans
- Increase datacenter scalability and agility

Conclusion

From the very beginning of the Internet, the DNS service was performing resolution of IP addresses, and it still does today. DNS is a very important component of the IP networks allowing users to access their applications, it is located at the intent of most IP communications. Specific usages and requirements have pushed network engineers to invent new protocols and devices in order to cover more functional needs like security.

Edge DNS GSLB functionality helps network engineers improve their network topologies in order to enhance the way users get access to their application. It allows to reduce design complexity, to handle easily specific situations, to limit investments and to simplify troubleshooting. Example usages can be found in the document: “Edge DNS GSLB Use Cases: Improving UX, DRP and Datacenter Agility”. Edge DNS GSLB is therefore the perfect companion to your load-balancing and multi-cloud strategy.

efficient iP®

REV: C-200810

Americas
EfficientIP Inc.
1 South Church Street
West Chester, PA 19382-USA
+1 888-228-4655

Europe
EfficientIP SAS
90 Boulevard National
92250 La Garenne Colombes-FRANCE
+33 1 75 84 88 98

Asia
EfficientIP PTE Ltd
60 Paya Lebar Road #11-47
Paya Lebar Square SINGAPORE 409051
+65 6678 7752