ANYCAST AS A LOAD BALANCING FEATURE

DNS TTL (Time To Live) is a value in a DNS record that determines how long DNS resolvers and browsers should cache the DNS information before requesting a fresh copy from the authoritative DNS servers.

When you visit a website, your device first needs to translate the domain name (like example.com) into an IP address. It does this by querying DNS servers. Once it gets an answer, your device or your ISP's DNS resolver will store (cache) this information for the duration specified by the TTL.

The TTL is measured in seconds. For example:

- A TTL of 3600 means the DNS record will be cached for 1 hour
- A TTL of 86400 means it will be cached for 24 hours

As you noted, TTL creates a trade-off:

- **Short TTLs** (minutes): Allow for faster DNS propagation when you make changes, which is useful during migrations or failovers. However, they increase the load on DNS servers due to more frequent lookups.
- **Long TTLs** (hours/days): Reduce DNS server load and can improve performance by reducing lookups, but make DNS changes take longer to propagate globally.

This is why TTL management is important for high-availability systems - if you need to quickly redirect traffic during an outage, a long TTL can indeed result in extended perceived downtime for users whose DNS resolvers have cached the old information.

2. What is Anycast and How Does It Work?

Anycast is a routing technique where multiple geographically distributed servers share the **same IP address**.

◆ **How Anycast Works:**

- When a user requests a resource (e.g., a website), the request is **automatically routed to the nearest server**.
- This is done by **BGP announcing the same IP address from multiple locations**.

- The internet's routing system then directs the request to the "closest" server (based on BGP metrics like AS path length, latency, or congestion).

🔹 **Example of Anycast in Action:** Imagine you have a globally distributed DNS service with servers in:

- New York
- London
- Tokyo

Each of these servers **advertises the same IP address using BGP**.

- A user from the **U.S.** will be routed to **New York**.
- A user from **Europe** will be routed to **London**.
- A user from **Japan** will be routed to **Tokyo**.

If the **New York** server goes down, BGP **automatically reroutes** U.S. traffic to London or Tokyo without user intervention.

Before Anycast, **proxy-based failover** was a common solution for handling failures.

🔹 **How Proxy-Based Failover Works:**

- A **proxy server** sits between the client and the backend servers.
- If the main backend fails, the proxy redirects traffic to a secondary server.
- However, proxies introduce:
  - **Latency:** Extra processing time due to rerouting.
  - **Loss of Client Identity:** Proxies often mask the original client's IP address.
  - **Complexity:** Additional infrastructure required.

Another advantage of having Load Balancers as Anycast peers is the reduced number of routing changes, because the Load Balancer combines multiple instances of a service into one VIP. That has been one of the concerns regarding Anycast deployments. Having the Load Balancer deal with service specific state healthchecks makes it possible to deploy Anycast not only for UDP based services, but also for TCP based

services – I guess this means that we dont have to advertise several paths to each separate server , we only have to advertise the load balancers and the actual routing is done by the load balancer

- **Special Subnet for Anycast**: They've set aside a specific section of their network addresses (a subnet) that's only used for Anycast virtual IP addresses (VIPs). This is like reserving a special neighborhood where only Anycast addresses can live.
- **Load Balancers Announce Routes**: The load balancers tell the routers, "Hey, I can handle traffic for this specific IP address" (that's the "/32 route advertisements" - a /32 means a single specific IP address).
- **Router Configuration**: The routers are set up to only accept these announcements if they're for addresses in that special Anycast subnet.
- **Protection Against Mistakes**: This creates a safety mechanism. If someone accidentally configures a system to announce routes for an IP address that's not in the special Anycast subnet, the routers will ignore it.
- **Preventing IP Takeover**: This stops accidental "takeovers" where a misconfigured system might try to claim an IP address that's already being used for something else, which could cause outages or routing problems.