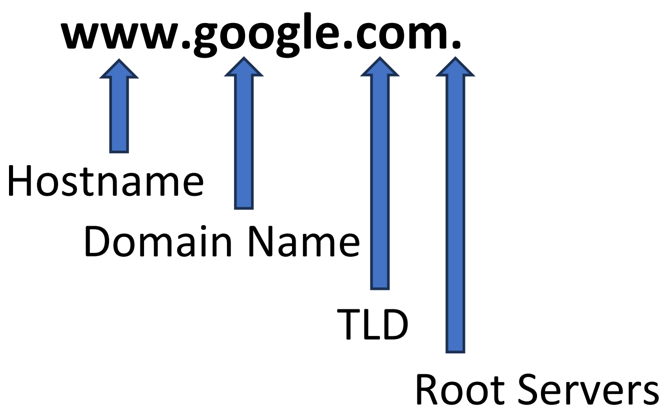
Setting up Simple Anycast DNS Recursive Resolver for an ISP

Kaung Khant Kyaw
Myanmar Internet Exchange

What is DNS?

- Resolve a fully qualified domain name (FQDN) to an IP Address IPs.
- Hierarchical and decentralized naming system.



Example FQDN : **www.google.com.**

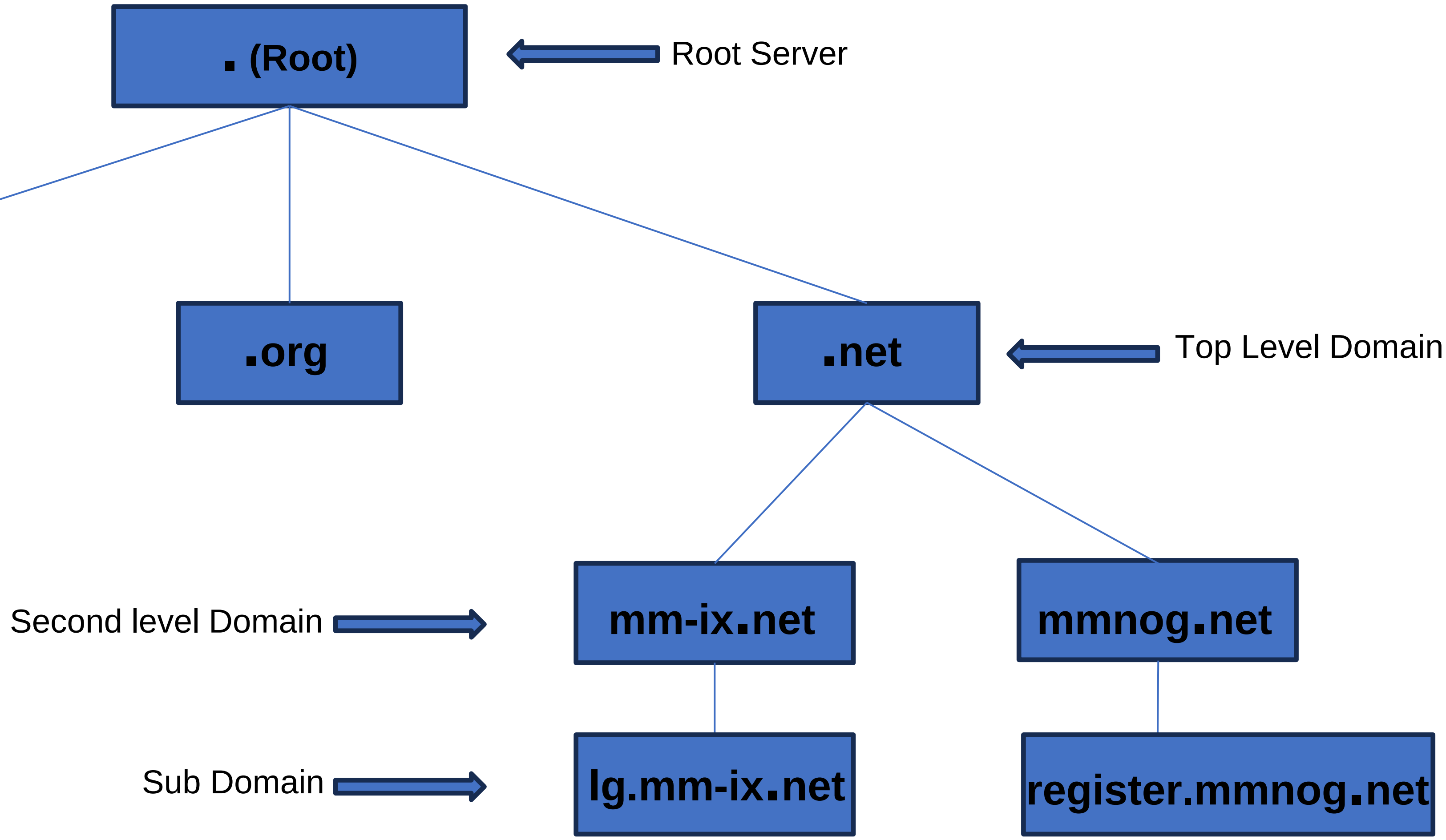
Hostname

Domain Name

TLD

Root Servers

Example of “DNS Hierarchy Structure”



What is a Recursive DNS Server?

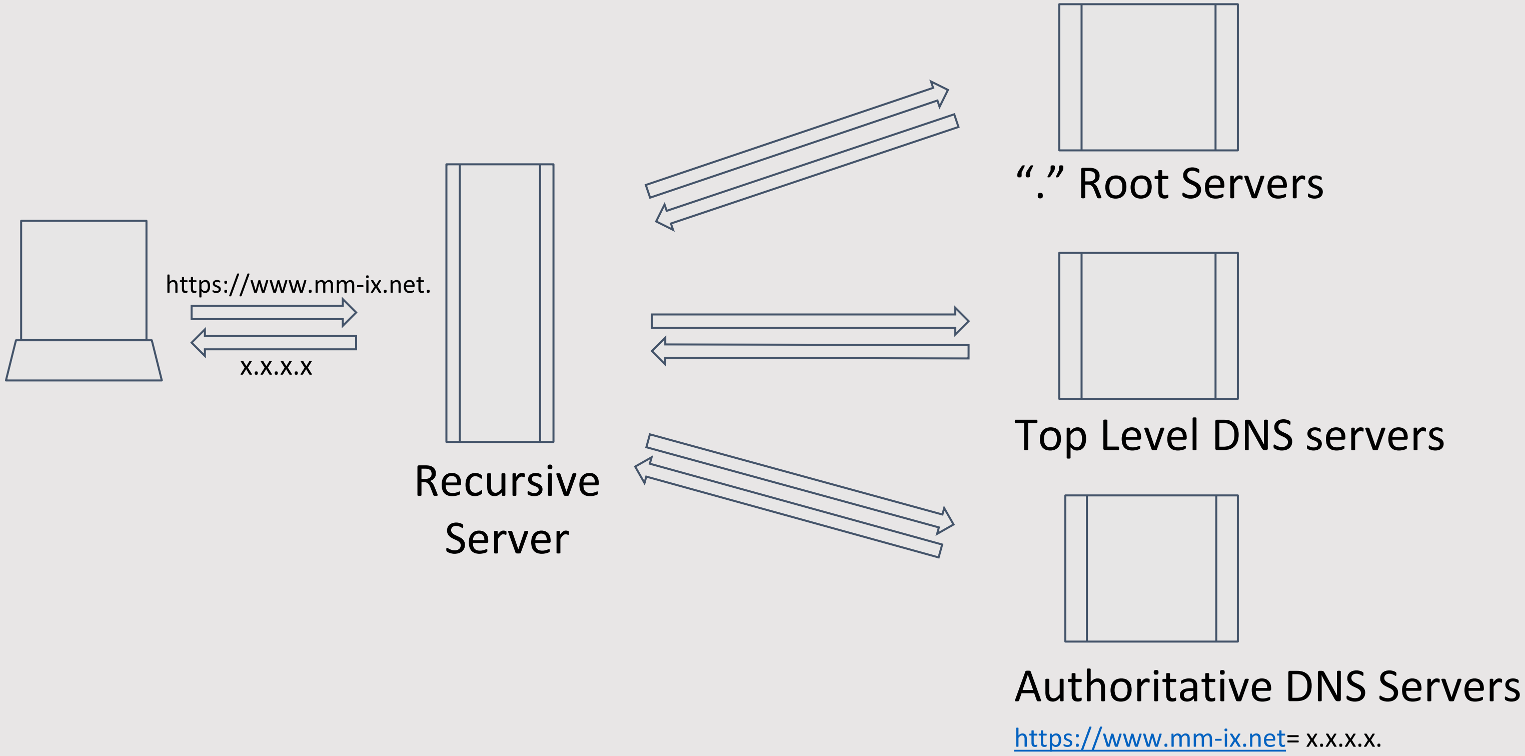
A recursive DNS server, also known as a recursive resolver or simply a resolver, is a fundamental component of the Domain Name System (DNS) infrastructure.

It’s is responsible for resolving the query. It starts by querying the root DNS servers, which point it to the appropriate Top-Level Domain (TLD) DNS server. The TLD server then directs the recursive server to the authoritative DNS server for the specific domain.

Recursive DNS Server's Mission:

- Is to efficiently and reliably resolve domain names to their corresponding IP addresses.
- It is emphasized that Name resolution, Caching, Query forwarding,...etc.
- Describe it as the essential intermediary that makes internet navigation seamless for users.

- **Here’s the example picture how “Recursive DNS Server” works:**



Type of Queries

Recursive Query

Is between DNS Client and its DNS Local Server.

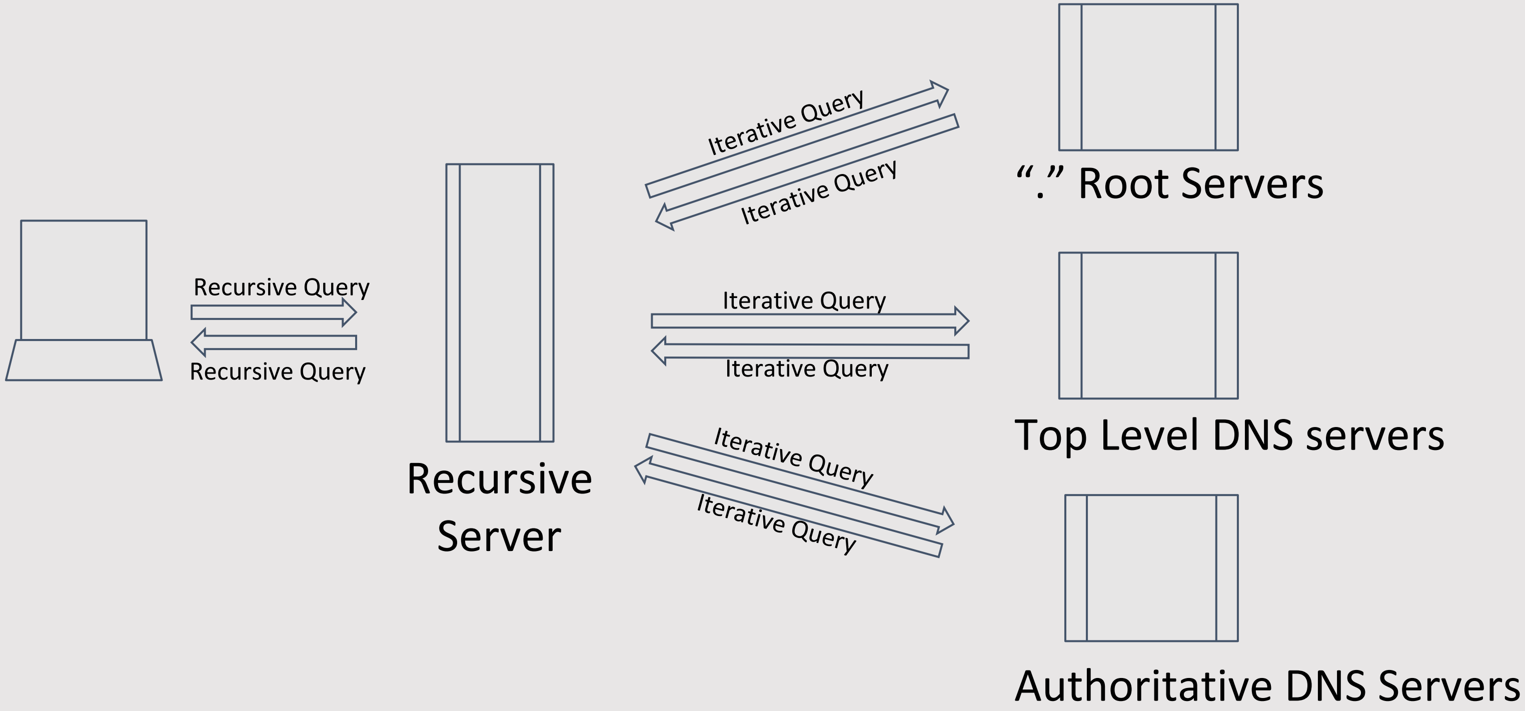
Demands a name resolution or the answer “It can not be found”

Iterative Query

Is between local DNS Server and other DNS servers.

Other DNS servers may provide a name resolution or a referral.

- **Here’s the example picture how “Recursive DNS Server” works:**



Advantages of
“ Local Recursive Resolver”

Efficient Name Resolution

Reduced Network Traffic

Improved User Experience

Load Balancing & Redundancy

...etc.

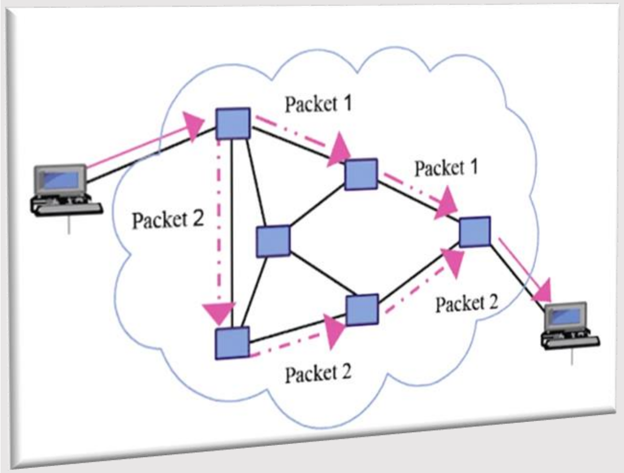
Different between
“Authoritative DNS Server” and “Recursive DNS Server”

	Authoritative DNS Server	Recursive DNS Server
Function	Provide official DNS records for a particular domain	Help resolve domain names to IPs by querying authoritative DNS servers.
Responsibility	Store and maintain DNS records for the domains	Don't store official records; they retrieve and cache info from authoritative servers.
Queries	respond to queries about specific domains with the accurate DNS records.	Issue queries to authoritative DNS servers to find the IP address associated with a domain.
Caching	don't typically cache records for other domains	Cache the results of queries to speed up future lookups.

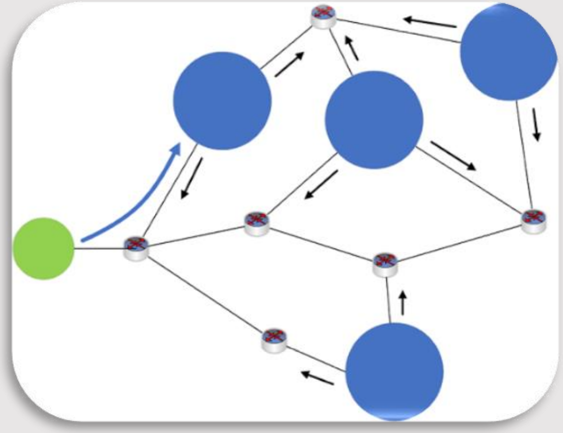
What is Anycast?

What is anycast? & Why Anycast?

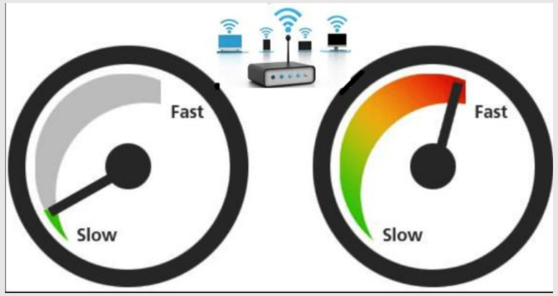
- Routing technique which routes packets to nearest destination.



- Reduces latency by directing traffic to the closest server.
- Provides redundancy and fault tolerance.

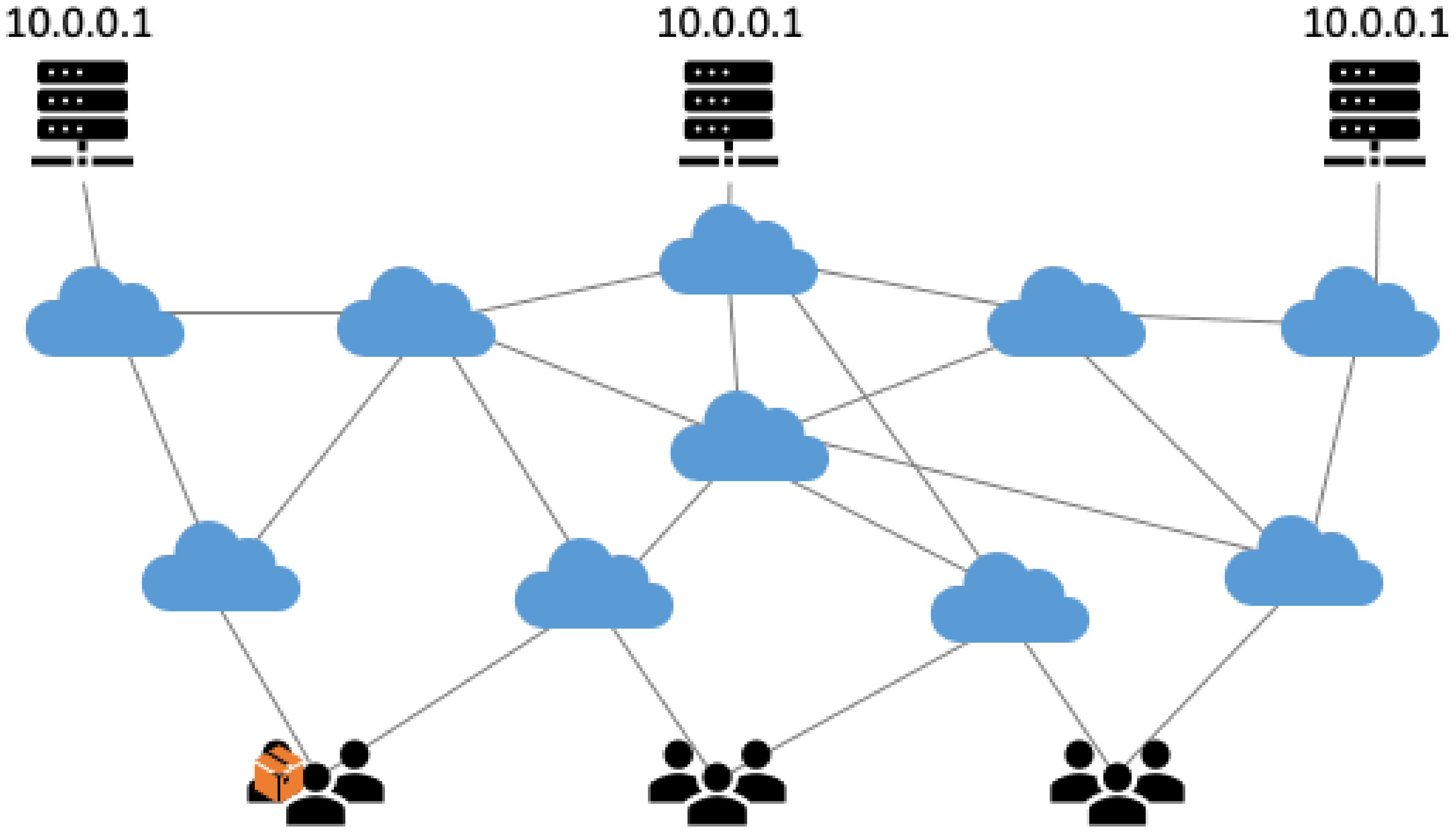


- Enhances performance and reliability.



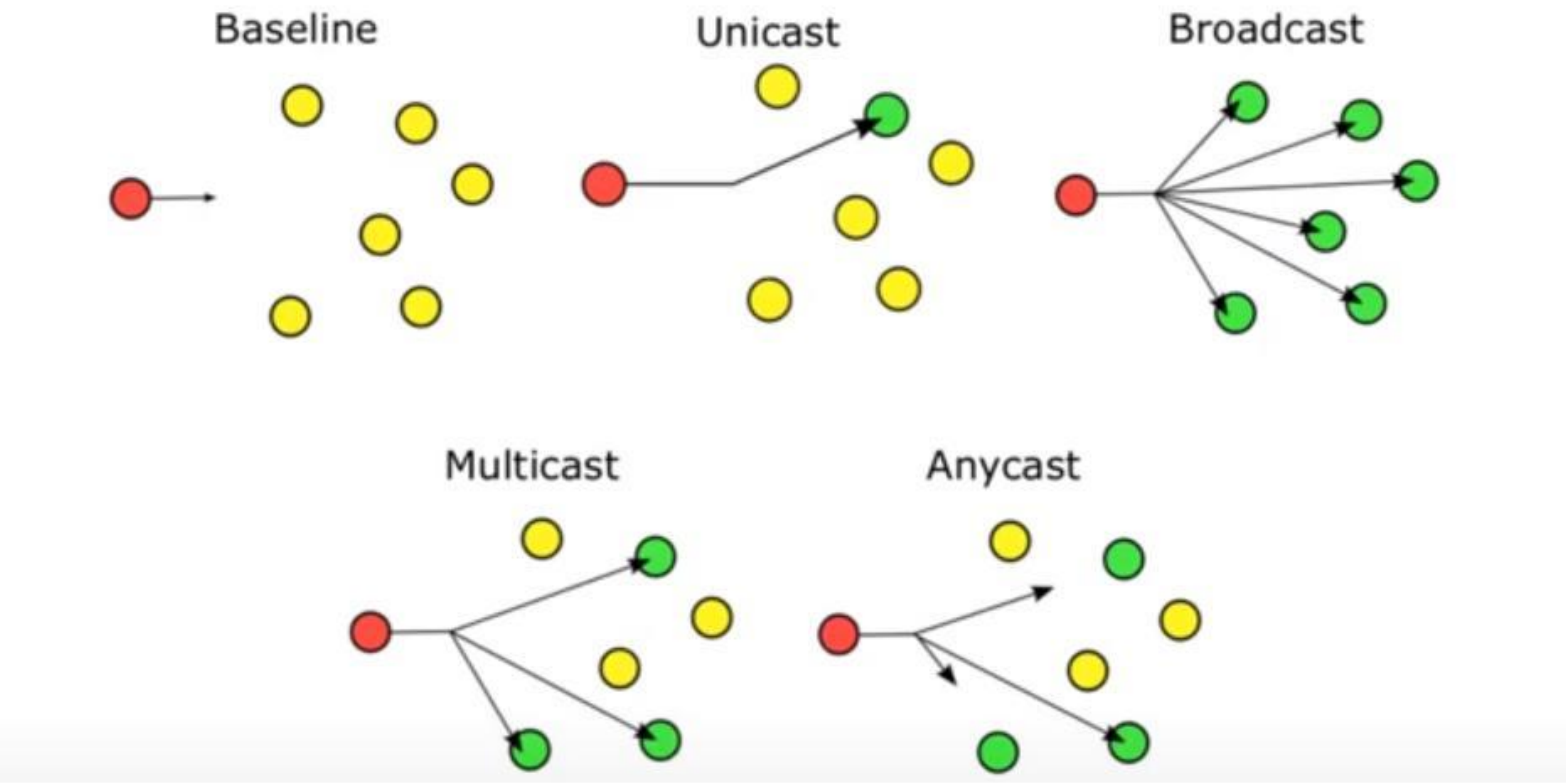
- Often used for DDoS mitigation.

- **The example of Anycast network routing!!!**



Reference from ***SIDN Labs***

Routing Schemes Compared



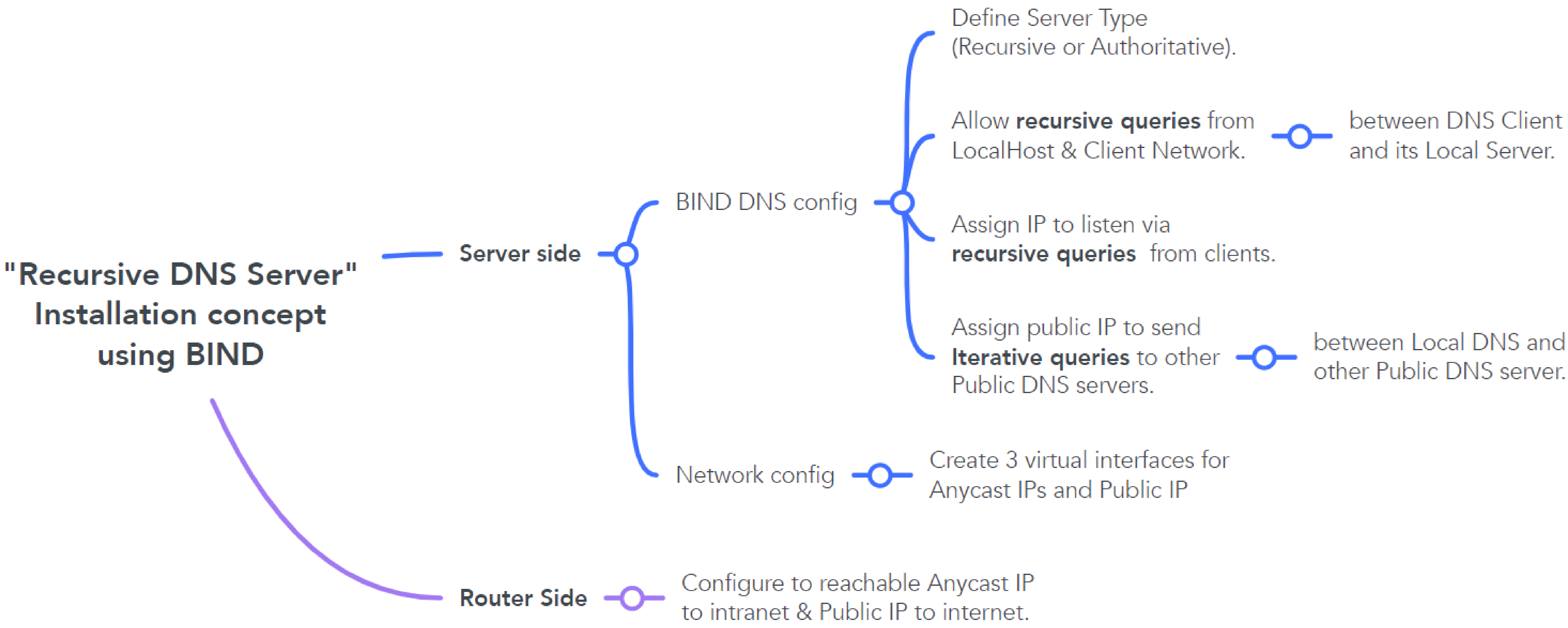
Reference from ***ISOC Anycast DNS training***

Advantages of running anycast on recursive DNS servers

- Running anycast on recursive DNS servers provides several advantages that enhance the performance, availability, and reliability of DNS services. Anycast is a network addressing and routing technique that allows multiple servers located in different geographic locations to share the same IP address.

  - ➢ Load Distribution
  - ➢ Proximity-based Routing
  - ➢ High Availability
  - ➢ Scalability
  - ➢ Cost Efficiency
  - ➢ Simplified Configuration
  - ➢ ……etc.

Recursive DNS Server Installation concept.



"Recursive DNS Server" Installation concept using BIND

Server side

BIND DNS config

Define Server Type (Recursive or Authoritative).

Allow **recursive queries** from LocalHost & Client Network.

between DNS Client and its Local Server.

Assign IP to listen via **recursive queries** from clients.

Assign public IP to send **Iterative queries** to other Public DNS servers.

between Local DNS and other Public DNS server.

Network config

Create 3 virtual interfaces for Anycast IPs and Public IP

Router Side

Configure to reachable Anycast IP to intranet & Public IP to internet.

Recursive DNS Server Installation concept.

recursion

- To define Authoritative server or Recursive DNS Server.

allow-query

- To allow who can query the domain to this server.

listen-on

- To listen client DNS queries

query-source

- To send query to another DNS server.

Recursive DNS Server Installation Example.

- Bind DNS Configuration

/etc/bind/named.conf.options

```
options {
        directory "/var/cache/bind";
        recursion yes; // allow recursive queries
        allow-query { localhost; 172.16.255.0/24; }; // allow queries from localhost and client network
        listen-on { localhost; 10.10.10.10; 11.11.11.11; }; // allow server IPV4 address to listen DNS queries from client
        query-source 103.103.1.1; //Specific IP address for DNS Server Outbound Query

        dnssec-validation auto;

        listen-on-v6 { any; };
};

logging{
        channel query logging {
                file "/var/log/named/query.log" versions 3 size 10m;
                severity debug 3;
                print-time yes;
                print-severity yes;
                print-category yes; };
        category queries {
                query logging; };
};
```

Recursive DNS Server Installation Example.

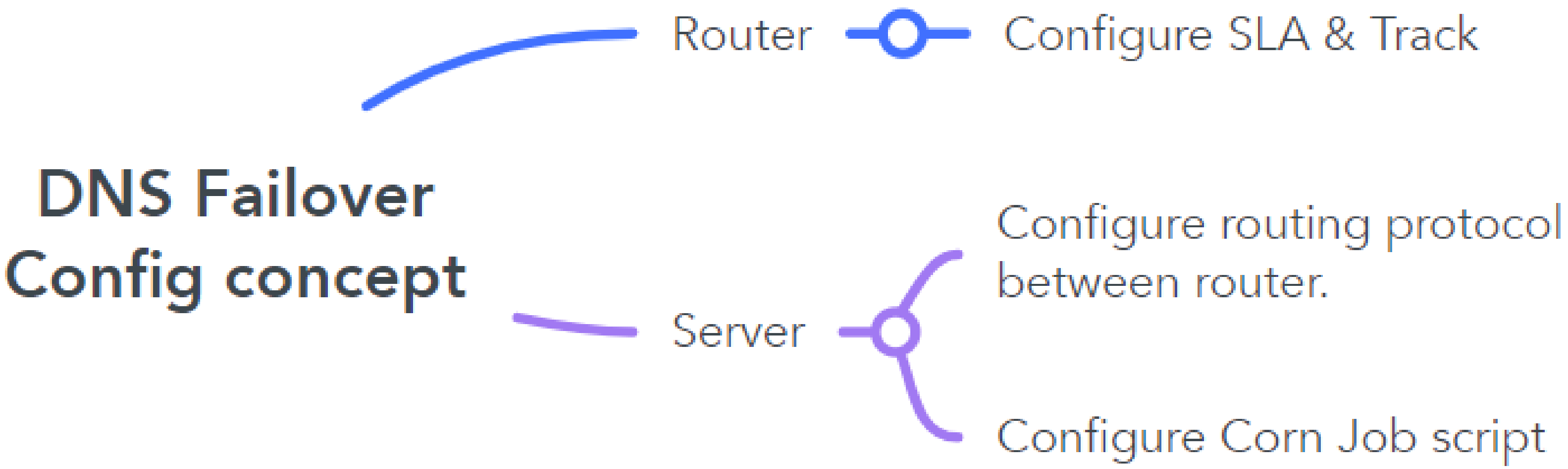
- Network Configuration

$ IP addr

```
ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fqcodel state UP group default qlen 1000
   link/ether 00:50:01:00:16:00 brd ff:ff:ff:ff:ff:ff
   inet 172.16.1.2/24 brd 172.16.1.255 scope global ens3
     valid_lft forever preferred_lft forever
   inet6 fe80::250:1ff:fe00:1600/64 scope link
     valid_lft forever preferred_lft forever
any10.vip: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
   link/ether 0e:f0:e8:74:2e:69 brd ff:ff:ff:ff:ff:ff
   inet 10.10.10.10/32 scope global any10.vip
     valid_lft forever preferred_lft forever
   inet6 fe80::cf0:e8ff:fe74:2e69/64 scope link
     valid_lft forever preferred_lft forever
any11.vip: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
   link/ether 3a:e7:76:ec:28:13 brd ff:ff:ff:ff:ff:ff
   inet 11.11.11.11/32 scope global any11.vip
     valid_lft forever preferred_lft forever
   inet6 fe80::38e7:76ff:feec:2813/64 scope link
     valid_lft forever preferred_lft forever
 public.vip: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000
   link/ether 22:94:a7:80:e7:e2 brd ff:ff:ff:ff:ff:ff
   inet 103.103.1.1/32 scope global public.vip
     valid_lft forever preferred_lft forever
   inet6 fe80::2094:a7ff:fe80:e7e2/64 scope link
     valid_lft forever preferred_lft forever
```

Failover Concept

Failover Configuration basic concept.



DNS Failover Config concept

Router

Configure SLA & Track

Server

Configure routing protocol between router.

Configure Corn Job script

Configuration Example. (Router-side)

#SLA Configuration

```
Router(config)# ip sla (operation number)
Router(config-ip-sla)#dns [hostname] name-server [DNS server address]
Router(config-ip-sla-dns)# frequency [ - ]
Router(config-ip-sla-dns)# threshold  [ - ]
Router(config-ip-sla-dns)# timeout [ - ]
```

#Track Configuration

```
Router# track [obj-No.] ip sla [entry No.] reachability
Router# delay up 1 down 1
```

#Configuring floating Static route with Track object

```
Router# ip route [DNS-server IP] [Netmask] [Next-Hop] track [obj-No.] [Metric]
```

Configuration Example. (Server-Side)

# Adding Cron Job Script

```
#!/bin/bash
while true; do
/usr/bin/dig @10.10.10.10 localhost. A +short
if [[ $? != 0 ]]; then
        echo "DNS service down"
        /etc/init.d/frr stop
else
        echo "DNS service up"
        /etc/init.d/frr enable
        /etc/init.d/frr start
fi
done
```

Thank you