1. **What is Dynamic Routing mode and difference between Global and Regional routing?**

**Dynamic routing** is a networking technique that provides optimal data routing. Unlike static routing, dynamic routing enables routers to select paths according to real-time logical network layout changes. In dynamic routing, the routing protocol operating on the router is responsible for the creation, maintenance and updating of the dynamic routing table. In static routing, all these jobs are manually done by the system administrator. **Dynamic routing** uses multiple algorithms and protocols. The most popular are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

The cost of routing is a critical factor for all organizations. The least-expensive routing technology is provided by dynamic routing, which automates table changes and provides the best paths for data transmission.

Typically, dynamic routing protocol operations can be explained as follows:

1. The router delivers and receives the routing messages on the router interfaces.

2.The routing messages and information are shared with other routers, which use the same routing protocol.

3.Routers swap the routing information to discover data about remote networks. 4.Whenever a router finds a change in topology, the routing protocol advertises this topology change to other routers.

**Dynamic routing** is easy to configure on large networks and is more intuitive at selecting the best route, detecting route changes, and discovering remote networks. However, because routers share updates, they consume more bandwidth than in static routing; the routers' CPUs and RAM may also face additional loads because of routing protocols. Finally, dynamic routing is less secure than static routing.

Let’s talk about **Global** and **Regional** **Routing**

When using regional dynamic routing mode, each Cloud Router in the VPC network only advertises subnet routes in the same region as the Cloud Router. When using global dynamic routing mode, each Cloud Router in the VPC network advertises **all subnet routes from all regions** of the VPC network.

2. **What is DNS policies and why do we need to Enable DNS Policies API in GCP?**

DNS is a hierarchical distributed database that lets you store IP addresses and other data and look them up by name. Cloud DNS lets you publish your zones and records in DNS without the burden of managing your own DNS servers and software. Cloud DNS offers both public zones and private managed DNS zones. A public zone is visible to the public internet, while a private zone is visible only from one or more Virtual Private Cloud (VPC) networks that you specify.

So first, lets understand what DNS Policies is so - **DNS Policy** used to allow primary and secondary **DNS servers** to respond to **DNS client** queries based on the geographical location of both the client and the resource to which the client is attempting to connect, providing the client with the **IP address** of the closest resource**. Google Cloud DNS** is a **high-performance**, **resilient**, **global Domain Name System (DNS) service**. As we know Application Programming Interface (API) is used to transportation of data without changing the application webpage or we can say endpoint. DNS Policy API is a service provided by Google in **GCP** to enable and apply the **DNS Policy** to the Virtual Machine Instances where the will assigned **IP address automatically.**

Cloud DNS uses anycast to serve your managed zones from multiple locations around the world for high availability. Requests are automatically routed to the nearest location, reducing latency and improving authoritative name lookup performance for your users.

Cloud DNS supports managed DNSSEC, protecting your domains from spoofing and cache poisoning attacks. When you use a validating resolver like Google Public DNS, DNSSEC provides strong authentication (but not encryption) of domain lookups. For more information about DNSSEC, see Managing DNSSEC configuration.

It is used when we create a Virtual machine Instance to get IP address according to the polices that we have applied in DNS Policies it mainly used in large organisation to auto scale the process for assignment of IP address to all the machines that are share the same DNS.

We only need to assign the DNS policy or different policies (we can create multiple DNS Servers and policies according to our use) and we can assign the DNS address to the machine where DNS starts it work and assign the IP Address to the connected machines.

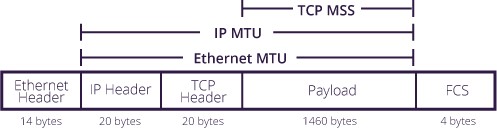
3. **What is MTU and why should we prefer 1460 rather 1500 as a value of MTU?**

**MTU** - MTU size is the maximum packet size that can be transmitted over your network. The MTU size simply determines the maximum packet size that can be transmitted over your network. MTU size is a property of the physical network interface and typically measures in bytes; the default size will be dependent on the type of network for example, the MTU size for Ethernet is 1500 bytes, whereas others will be larger, and others will be smaller If the messages transmitted are larger than the MTU size, they will be divided into smaller packets before being sent.

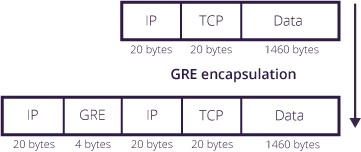
So why should we prefer small MTU value rather then 1500 so answered is

* The larger the MTU size is, the more data that can fit into fewer packets. This generally results in a faster and more efficient transmission of data across your network.
* On the other hand, if an error occurs, the packet will take longer to re-transmit. Also, more data within fewer packets can result in the packets being bloated.
* Larger packets are more likely to suffer from delays and even corruption. A greater MTU size can also increase latency, whereas setting it to a smaller number can help improve the overall latency.

So why we use MTU 1460? So, answer is MTU is built from payload (also referred as data) and the TCP and the IP header, 20 bytes each. The total value of the IP and the TCP header is 40 bytes and mandatory for each packet, which leaves us 1460 bytes for our data.



If we use MTU = 1500 then the total size of this kind of packet will be 1524 bytes, exceeding the 1500 bytes MTU value. The “data” size in this packet is 1460, but we can and should decrease it to make sure the total size will be 1500 bytes or less. And this is where TCP MSS comes into the picture.



TCP MSS, the maximum segment size, is a parameter of the options field of the TCP header that specifies the largest amount of data, specified in bytes, that a computer or communications device can receive in a single TCP segment. It does not include the TCP header or the IP header. This value will dictate the maximum size of the “data” part of the packet. In the following case for the GRE tunnel, we will set the TCP MASS value to be 1436 or lower, while the default size is 1460.

The MSS announcement (often mistakenly called a negotiation) is sent during the three-way handshake by both sides, saying: “I can accept TCP segments up to size x”. The size (x) may be larger or smaller than the default. The MSS can be used completely independently in each direction of data flow.

Since the end device will not always know about high level protocols that will be added to this packet along the way, like GRE packets for example, it won’t usually adjust the TCP MSS value. As a result, the network devices have the option to rewrite the value of TCP MSS packets that are processed through them. For example, in a Cisco Router the command “ip tcp mss-adjust 1436” in the interface level will rewrite the value of the TCP MSS of any SYN packet that will go via this interface.

I guess everything I explained properly if still you need any correction or need to add some point please get back to me.

# PTO