Topic 5D: Compare Network Configuration Concepts:

Dynamic Host Configuration Protocol:

When a device is assigned a **static IP address manually**, there is a risk of **human error**. The installer might **accidentally enter the wrong IP address**, **use a duplicate IP** (causing conflicts), or **set the wrong subnet mask**, which could prevent communication with other devices. Additionally, if the **network changes**, each affected device must be **manually reconfigured**, which is time-consuming.

To solve these issues, networks use a **DHCP (Dynamic Host Configuration Protocol) server**, which **automatically assigns the correct IP address, subnet mask, and other settings** to any device that connects to the network. Similarly, **DHCP ensures that every device on a network gets a unique and correct IP configuration without manual setup, reducing mistakes and saving time**.

DHCP Scope:

A **DHCP Scope** is the **range of IP addresses** that a **DHCP server** can assign to devices on a network. It ensures that devices **automatically receive a valid IP address** without conflicts. However, the **scope should not include IP addresses that are assigned manually (static IPs)**, such as the router's IP address or servers that need fixed addresses.

- **Example:**

In a **home or small office (SOHO) network**, the router has a **static IP of 192.168.0.1**, which must **not** be included in the DHCP scope. If the DHCP scope is set to **192.168.0.100 to 192.168.0.199**, the router can assign **100 different IPs** to devices like laptops, phones, and printers **dynamically** as they connect.

DHCP Leases:

A **DHCP lease** is the **temporary assignment of an IP address** from a **DHCP server** to a device (DHCP client). Instead of permanently assigning IPs, the **DHCP server "leases" the address** for a certain period, after which the device must **renew the lease** or request a new IP. This process **ensures efficient IP address management** and prevents conflicts.

- **How a Device Gets an IP from DHCP (Step-by-Step Process)**
  1. **DHCPDISCOVER** – The device (DHCP client) **broadcasts** a request looking for a DHCP server.
  2. **DHCPOFFER** – The DHCP server **replies** with an available IP address and other settings (subnet mask, gateway, DNS).
  3. **DHCPREQUEST** – The device **requests** to use the offered IP.
  4. **DHCPACK** – The server confirms the assignment, and the device **starts using the IP**.

✓ **The client then checks** if the IP is already in use (by sending an **ARP request**). If no one responds, the client **keeps the IP**; otherwise, it requests a new one.

- **What Happens When the Lease Expires?**
  - Before the lease expires, the **device tries to renew** the IP with a **DHCPREQUEST**.
  - If the **renewal fails** (e.g., the DHCP server is down), the device **releases** the IP and **starts the discovery process again**.
  - If the network changes (new gateway, DNS, etc.), the DHCP server updates settings automatically, and devices **apply the changes without manual reconfiguration**.

Why Does a DHCP Server Need a Static IP?

A **DHCP client (device requesting an IP)** does **not** need to know the **DHCP server's address** in advance. Instead, it **broadcasts** a request on the network to find a DHCP server. Because of this, a DHCP server **must have a static IP address** so that clients **always** know where to send their requests.

◆ **Real-Life Example:**

Imagine you walk into a **restaurant** 🍽️ and want to order food. You **don't need to know the exact name of the waiter (DHCP server)**—you just **call out** for service (broadcast request), and a waiter **comes to take your order**. However, if the **waiter kept changing jobs (dynamic IP)**, customers wouldn't know who to call. To avoid confusion, the waiter **always works at the same station (static IP), making service efficient**. Similarly, if a **DHCP server had a changing (dynamic) IP**, devices wouldn't know where to **renew their leases** or request new IP addresses, causing network disruptions. That's why **DHCP servers are always assigned a static IP** for stability.

DHCP Reservations:

A **DHCP reservation** ensures that **certain devices always get the same IP address** without needing manual configuration. Instead of setting a **static IP** on the device itself, the **DHCP server is configured to "reserve" a specific IP for that device's MAC address**. This makes managing **servers, routers, printers, and other network devices** easier because their IPs remain **consistent**, but they still benefit from **automatic DHCP management**.

◆ **How DHCP Reservations Work**

1. A device (e.g., a printer) **sends a DHCP request** to get an IP.
2. The **DHCP server checks its reservation list** for the device's **MAC address**.
3. If the MAC address is **on the list**, the DHCP server **assigns the reserved IP** to that device.
4. The device always **receives the same IP** whenever it connects.

Some operating systems send a different unique identifier than a MAC address by default. The identification method should be configured appropriately on the client so that the server has the correct information.

Domain Name System:

An **IP address** is a numerical way to identify devices on a network, but it is **difficult for people to remember**. Instead, networks use **hostnames** combined with **domain names** to create **human-readable addresses** called **Fully Qualified Domain Names (FQDNs)**. An **FQDN** is a complete domain name that uniquely identifies a device on the internet.

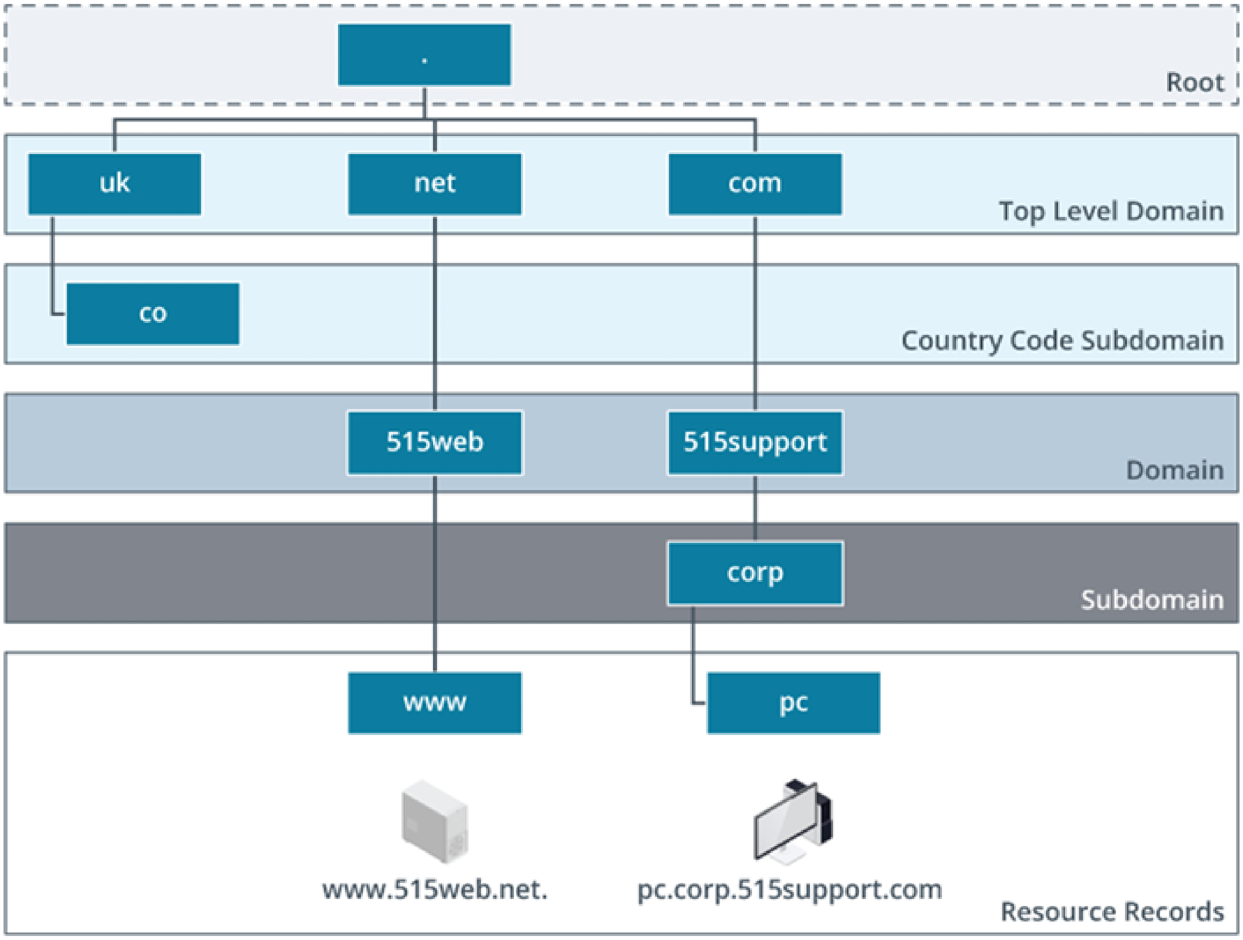
Example of an FQDN: pc.corp.515support.com

- **pc** is the hostname of the device.
- **corp** is a subdomain used to organize different parts of a business.
- **515support** is the domain name registered under **.com**.
- **.com** is the **Top-Level Domain (TLD)** that categorizes websites.

How DNS Works

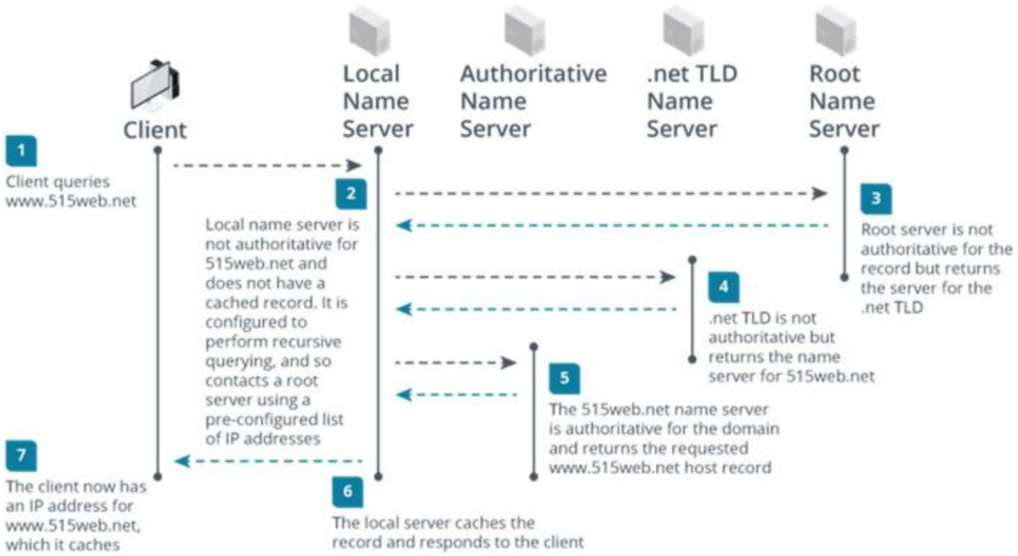
The **Domain Name System (DNS)** translates **FQDNs into IP addresses**, allowing users to access websites without remembering complex numerical IPs. DNS is structured as a **hierarchical system**, with **root servers** at the top, followed by **TLDs (.com, .org, .net, etc.)**, and then individual domains and subdomains.

Real-Life Example: Imagine looking up a person's name in a **phone book** instead of remembering their phone number. DNS acts like this **phone book**, converting names like www.google.com into an IP address that computers understand.



DNS hierarchy. (Images © 123RF.com.)

DNS Queries:



1 Client queries www.515web.net

2 Local name server is not authoritative for 515web.net and does not have a cached record. It is configured to perform recursive querying, and so contacts a root server using a pre-configured list of IP addresses

3 Root server is not authoritative for the record but returns the server for the .net TLD

4 .net TLD is not authoritative but returns the name server for 515web.net

5 The 515web.net name server is authoritative for the domain and returns the requested www.515web.net host record

6 The local server caches the record and responds to the client

7 The client now has an IP address for www.515web.net, which it caches

DNS name resolution process. (Images © 123RF.com.)

When a user types a **Fully Qualified Domain Name (FQDN)** like www.515web.net into a web browser, the **DNS system** translates it into an **IP address** so the computer knows where to connect. This process is called **DNS resolution** and happens in multiple steps.

How DNS Name Resolution Works

1. **Client Query** – The user enters www.515web.net into a web browser. The device first **checks its local DNS cache** for a saved IP address. If no record exists, it contacts the **local DNS server** (set in the TCP/IP settings).
2. **Local Name Server** – If the local DNS server does not have the IP address, it sends a query to the **Root Name Server** to find the correct DNS path.
3. **Root Name Server** – The root server does not know the specific IP for www.515web.net, but it directs the query to the **TLD (Top-Level Domain) name server** for **.net**.
4. **TLD Name Server** – The **.net name server** does not have the exact IP but knows which **authoritative name server** handles **515web.net** and directs the query there.
5. **Authoritative Name Server** – The authoritative name server for **515web.net** has the actual **IP address** for www.515web.net and returns it to the local DNS server.
6. **Response to Client** – The local DNS server **caches** the response and forwards the IP address to the client.
7. **Client Connects** – The client now has the correct IP and can connect to www.515web.net without repeating the process unless the cache expires.

DNS Record Types and Their Functions:

A **DNS server** is responsible for storing and managing **resource records** that map domain names to IP addresses. These records help DNS resolve queries and provide information about network services. DNS records can be **manually configured (static)** or **automatically updated (dynamic)** based on network changes.

Common DNS Record Types

A Record (Address Record) – Maps a **domain name** to an **IPv4 address** (e.g., www.example.com → **192.168.1.1**).

AAAA Record (Quad-A Record) – Maps a **domain name** to an **IPv6 address** (e.g., www.example.com → **2001:db8::1**).

CNAME Record (Canonical Name) – Creates an **alias** for another domain name. For example, **blog.example.com** can point to **example.com** instead of having a separate A record.

MX Record (Mail Exchange) – Specifies the **email server** responsible for receiving mail for a domain (e.g., **mail.example.com**).

NS Record (Name Server) – Identifies the **authoritative DNS servers** for a domain, directing queries to the correct name servers.

PTR Record (Pointer Record) – Used for **reverse DNS lookups**, mapping an IP address back to a domain name.

SRV Record (Service Record) – Defines the **location of specific services**, such as VoIP or LDAP servers.

TXT Record (Text Record) – Stores **text-based information** for domains, often used for security purposes like SPF (Sender Policy Framework) and domain verification.

DNS Spam Management:

A **TXT record** in DNS is used to store **free-form text information** that helps with various network functions. One of its most common uses is **email verification and spam prevention**. A domain can have multiple **TXT records**, mainly used to **verify email servers** and **block spam or spoofed messages**.

Imagine you own a **restaurant** and want to ensure that only your **trusted employees** can take reservations on behalf of your business. However, some **scammers** might try to **pretend to be your restaurant** and take fake bookings to damage your reputation. To prevent this, you set up strict **verification rules** so that customers know they are speaking with a **real employee** and not a fraudster.

In the **email world**, businesses and organizations use **DNS TXT records** like **SPF, DKIM, and DMARC** to **verify email legitimacy** and **prevent email scams (spoofing and phishing attacks)**. Here's how each one works using the restaurant example:

SPF (Sender Policy Framework) – Employee List
Imagine you create a **list of employees** who are **allowed to take reservations**. If a customer calls, they can **check the official employee list** to confirm they are speaking to a real staff member.

✓ In email terms, **SPF is a TXT record in DNS that lists the authorized mail servers** that can send emails on behalf of your domain. If an email comes from an **unauthorized server**, it is flagged as **suspicious** or **rejected**.

🔹 **Example:** If your company's domain is **@yourbusiness.com**, SPF ensures that only your **official mail server** (e.g., Gmail or Microsoft Outlook) can send emails using that domain. If an attacker tries to send a fake email from **@yourbusiness.com** using their own server, SPF will block it.

DKIM (DomainKeys Identified Mail) – Signature Verification
In the restaurant, employees are required to **sign each reservation receipt** before handing it to customers. If someone receives a receipt without a signature or with a fake signature, they know it's not **legitimate**.

✓ In email terms, **DKIM adds a unique cryptographic signature to every outgoing email**. The recipient's mail server **checks this signature** to ensure that the email was **not altered or forged** during transmission.

🔹 **Example:** If your business sends an invoice to a client, DKIM ensures that no one **modifies the amount or bank details** in the email before it reaches the recipient.

DMARC (Domain-Based Message Authentication, Reporting & Conformance) – Security Rules & Penalties
To fully protect your restaurant, you create **a policy** stating that if someone **fails to follow the employee list (SPF) or signature rule (DKIM),** they should be **reported** and their fake reservations should be **rejected**.

✓ In email terms, **DMARC tells email servers what to do if SPF and DKIM checks fail**. It can instruct them to **reject, quarantine, or monitor suspicious emails** to prevent phishing attacks.

🔹 **Example:** If a hacker tries to send a fake email pretending to be **your CEO**, DMARC tells the recipient's mail server to **reject the email outright or mark it as spam** instead of delivering it.

Real-World Email Scam Prevention Example
Imagine you get an email from **bank@securebank.com** asking for your password. How do you know it's really from your bank and not a scammer?

✓ **SPF checks if the email came from an authorized bank mail server.**

✓ **DKIM verifies that the email wasn't altered in transit.**

✓ **DMARC enforces security policies, ensuring that fake emails are rejected.**

If any of these checks fail, the email is **flagged as spam or rejected**, protecting you from phishing attacks.

Virtual Lans:

In a **basic network**, all devices connected to the same switch can communicate freely. This setup works fine for **small networks**, but in **large enterprise environments**, having hundreds or thousands of devices in the **same broadcast domain** can cause **network congestion and performance issues**. **VLANs (Virtual Local Area Networks)** help solve this by logically **dividing a switch into multiple isolated networks**, improving performance and security. Traffic passing between VLANs can easily be filtered and monitored to ensure it meets security policies.

How VLANs Work

A **VLAN is created by assigning switch ports** to a specific VLAN **ID** (ranging from **2 to 4094**). For example:

- **Ports 1-10** can be assigned to **VLAN 10 (Sales Department)**
- **Ports 11-20** can be assigned to **VLAN 20 (IT Department)**
- **Ports 21-30** can be assigned to **VLAN 30 (Guest Network)**

Even though all these devices **connect to the same physical switch**, they **cannot communicate** with devices in **other VLANs** unless a **router is used** for inter-VLAN communication.

Real-Life Example: Office Building Floors 🏢

Imagine a **corporate office** with three departments:

- **Sales on the 1st floor**
- **IT on the 2nd floor**
- **Guests in the lobby**

If everyone could freely **walk into any floor** (like an unmanaged switch), it could create **security risks and confusion**. Instead, the company **restricts access** using **keycards (VLAN IDs)**:

- Employees from **Sales can only access their floor (VLAN 10)**
- IT staff **can access their floor but not the guest area (VLAN 20)**
- Guests **can only use the lobby Wi-Fi (VLAN 30)**

If **Sales wants to communicate with IT**, they must go through **reception (a router)**, which controls and filters communication between departments.

Benefits of VLANs

- **Improves Network Performance** – Reduces **broadcast traffic** by isolating devices.
- **Enhances Security** – Prevents **unauthorized access** between VLANs.
- **Easier Traffic Management** – Separates **VoIP, guest networks, and sensitive data** for better **quality of service (QoS)**.

What is Broadcast domain?

A **broadcast domain** is a part of a network where **any device can send a broadcast message to all other devices** without needing a router. In simpler terms, it is a group of devices that **receive each other's network broadcasts**.

Imagine a **large open office** where **anyone can shout,** and **everyone hears them**—this is a **single broadcast domain**. If **too many people talk at once**, it becomes **noisy and hard to focus** (network congestion). Now, imagine the office is **divided into separate rooms** (like VLANs or subnets). People in **one room** (broadcast domain) **only hear each other**, but they need to **use a receptionist (router)** to talk to another room.

- A **broadcast domain** is where **all devices receive each other's broadcasts**.
- **Switches do NOT break broadcast domains**, but **routers and VLANs do**.
- **Too many devices in a broadcast domain** can slow down a network.
- **VLANs divide a switch into multiple broadcast domains**, improving efficiency.

Virtual Private Networks:

A **VPN (Virtual Private Network)** allows a user to **remotely connect to a private network** over the **public internet** as if they were physically present at the location. Instead of plugging into a local switch or Wi-Fi, a VPN user **connects through a remote access server** that authenticates the connection and establishes a **secure tunnel**. It uses special connection protocols and encryption technology to ensure that the tunnel is protected against snooping and that the user is properly authenticated.

How a VPN Works

1. The remote user **connects to the Internet** from any location.
2. The user starts a **VPN connection** to a company's VPN server.
3. The **VPN server authenticates** the user and establishes a **secure, encrypted tunnel**.
4. Once connected, the user **can access internal network resources**, such as company files, printers, or internal websites, just like being physically present in the office.

Real-Life Example: Remote Work

Imagine an employee working **from home** needing to access files stored on the company's network. Instead of traveling to the office, they use a **VPN connection** to securely log into the corporate network. Even though they are at home, the VPN makes their device **act like it's inside the company network**, allowing access to internal resources.

Types of VPNs

- **Remote Access VPN** – Used by employees working from home or while traveling to securely connect to their office network.
- **Site-to-Site VPN** – Used by businesses to connect **branch offices** securely over the internet.

Why Use a VPN?

- **Security** – Encrypts data, preventing hackers from intercepting sensitive information.
- **Privacy** – Hides the user's IP address and online activity from public networks.
- **Access Control** – Allows authorized users to securely connect to private networks from anywhere.

Some Questions and Answers:

1. **You need to ensure that a print device receives the same IP address when connecting to the network. What value do you need to configure on the DHCP server to enable a reservation?**
   The **MAC address** of the print device needs to be configured on the **DHCP server** along with the **specific IP address** to assign. This ensures the printer always gets the same IP when connecting.

2. **True or false? A top-level domain such as .com represents the top of the DNS hierarchy.**
   False. The **DNS hierarchy starts with root servers**, which are at the highest level. The root is represented by a **trailing dot (.)** in an FQDN, but this is often omitted in regular use.

3. **You are advising another technician about typical DNS configuration. The technician thinks that the name server hosting the 515support domain resource records on the Internet should be configured as the primary DNS server entry in the IP configuration of local clients. Why is this unlikely to be the case?**
   A **name server** for a specific domain (like **515support.com**) **only resolves queries for that domain**. A **local DNS server** listed in a client's configuration needs to **resolve queries for all domains** by contacting different name servers as needed. Mixing both roles on the same server is possible but is **not common due to performance and security reasons**.

4. **What type of value would you expect a query for an AAAA resource record to return?**
   An **IPv6 address**. AAAA records map domain names to **IPv6 addresses**, similar to how A records map to IPv4.

5. **What type of TXT record uses cryptography to help recipient servers reject spoofed messages and spam?**
   DomainKeys Identified Mail (DKIM). DKIM uses cryptographic signatures to verify that an email **was not altered** in transit and comes from an authorized sender.

6. **Which network configuration technology can be configured on switches to divide a local network into multiple broadcast domain segments?**
   Virtual LAN (VLAN). VLANs allow switches to **logically separate devices** into different network segments, improving **security and network performance**.