RNS INSTITUTE OF TECHNOLOGY

An Autonomous Institute under VTU
Accredited with NAAC A+ Grade

PROJECT

DHCP & DNS SETUP

Prepared By

MANOJ D

&

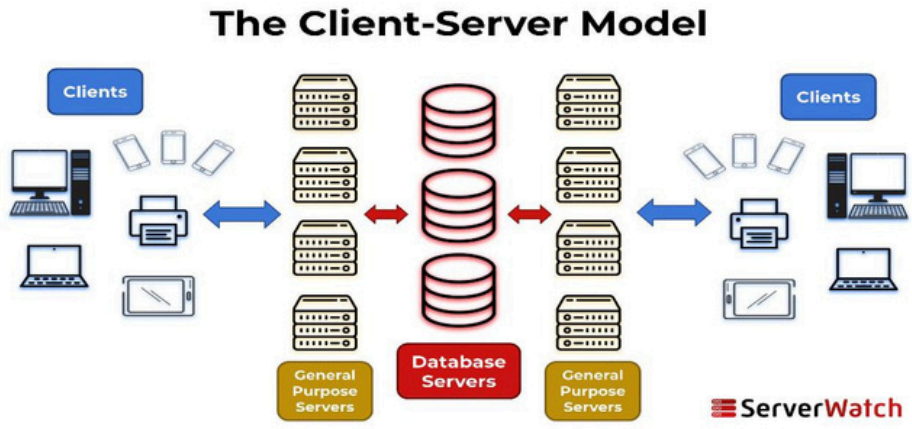
RAGHAV G

Client-Server Model (DHCP & DNS Setup)

Client-Server Model

.Introduction

With the advancement in technology, Web is becoming very much more important in our daily lives, in which virtually everything we do nowadays involves the use of web. More so, the application of Web is not limited to computers but it is opened to different kinds of intelligent digital devices, for example the mobile ones. Also, the architecture of the Web is the Client-Server model, in which as a result the communication between the client and server is the first thing we should be concerned about . Client/server system has increasingly minimized application development time by dividing functions of sharing information into both the client and server. The client is the requester while the server is the provider of service. In most client-server environment, the data processing is handled by the server, and the results are returned to the clients, which is made to speed up the rate of performance .For example, in a workstation, a printer can be attached to a computer (representing the clients) while other computers sharing from it are the server



How it Works

1. Client Request: A client, such as a web browser or an application on a user's computer, initiates a request for a service.

2. Network Communication:

This request is sent over a network, using protocols like HTTP/HTTPS for web applications or TCP/IP for general communication.

3. Server Response:

The server receives the request, performs the necessary processing, and then sends a response back to the client.

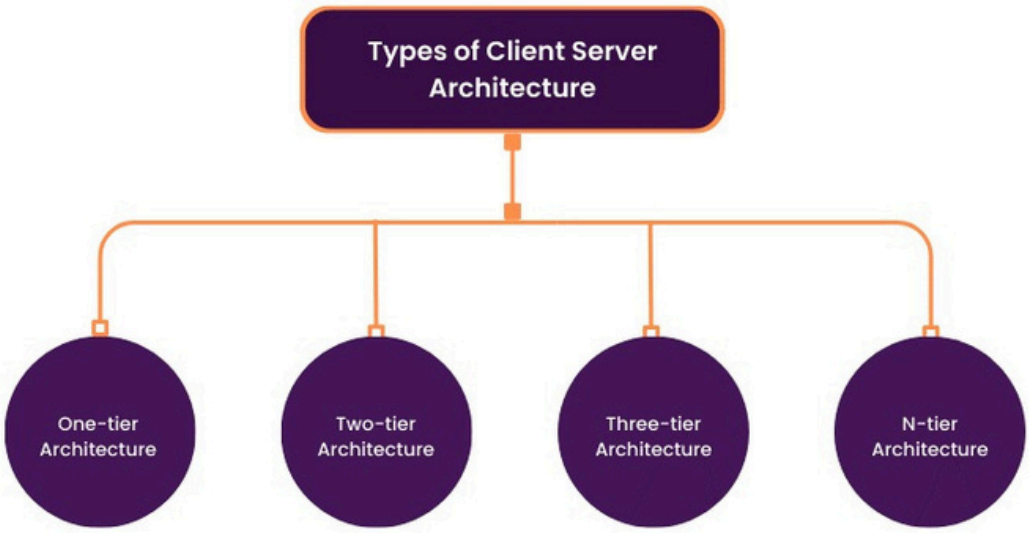
4. Data Exchange:

Information is exchanged using APIs (Application Programming Interfaces), which actas a contractdefining how theclientcalls the server and how the server responds.

Client-Server Systems Architecture

Client-server architecture is usually made up of the; application server, database server and PC.

The primary types of client-server architectures are 1-Tier, 2-Tier, 3-Tier, and N-Tier, differentiated by the number of logical and physical layers that separate presentation, business logic, and data responsibilities. 1-Tier places all components on a single system, while 2-Tier separates the client (presentation) from the server (data), and 3-Tier introduces a middle "application" layer for business logic. N-Tier architecture generalizes the 3-Tier model into an arbitrary number of layers for complex, scalable applications.

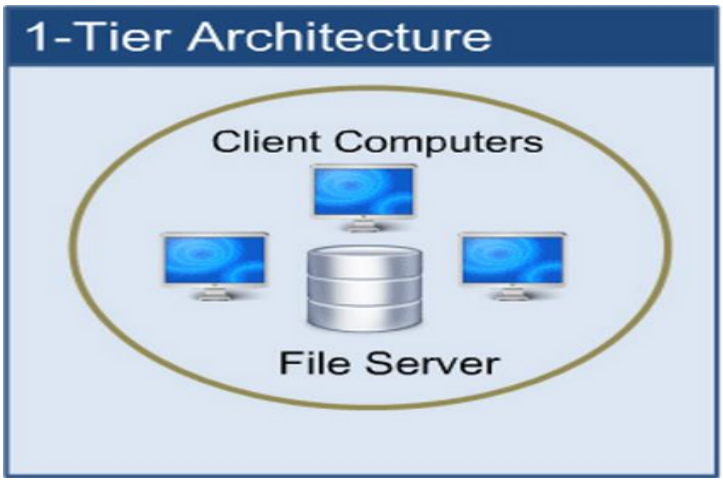


1-Tier Architecture

Description: All components—the presentation (user interface), business logic, and data—reside on a single machine.

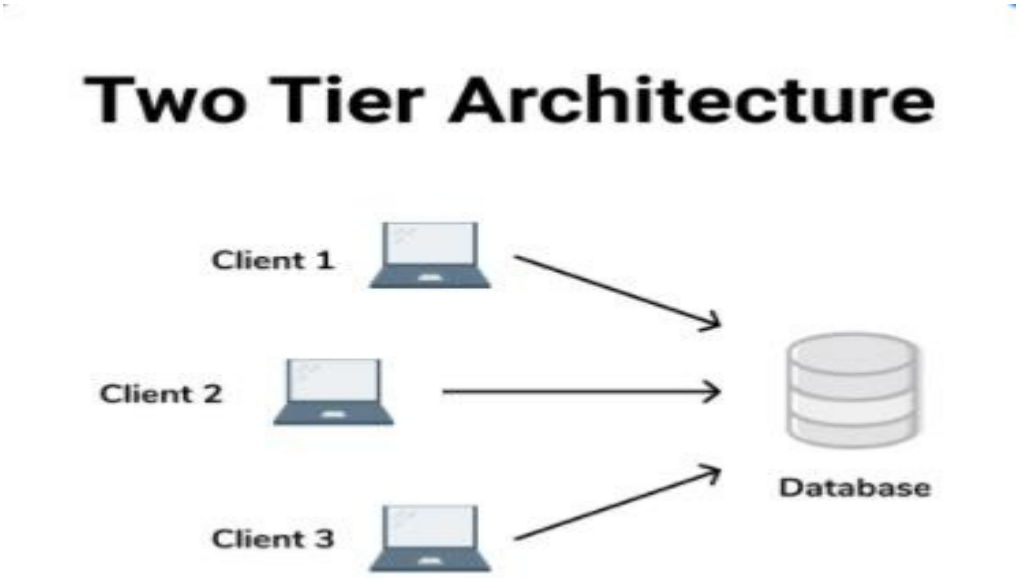
Characteristics: Simple, affordable, and requires no network traffic.

Example: A simple desktop application that stores and processes its data entirely on the same device.



2-tier client-server system architecture: This is an architecture which involves only the Database server and a client PC. In 2-tier architecture, the

users will run applications on their PC (Client), which connects through a network to the server. The client application runs both the coding and business logic, and then displays output to the user. It is also called thick client. -It is considered when the client has access to the database directly without involving any intermediary. -It is also used to perform application logic whereby the application code will be assigned to each of the client in the workstation.



3-tier client-server system architecture: This architecture involves the client PC,Database server and Application server. 3-tier architecture can be extended toN-tier whereby it involves more application servers.

Inthis architecture, the client contains presentation logic only, whereby less resources and less coding are needed by the client.

It supports one server being in charge of many clients and provides more resources in the server . It involves an intermediary (Application server) also known as middleware.

> Middleware: The 3-tier architecture involves an application server which serves as a middleware between the client PC and database server. The middleware tier is separate software running on a separate machine and performs application logic .

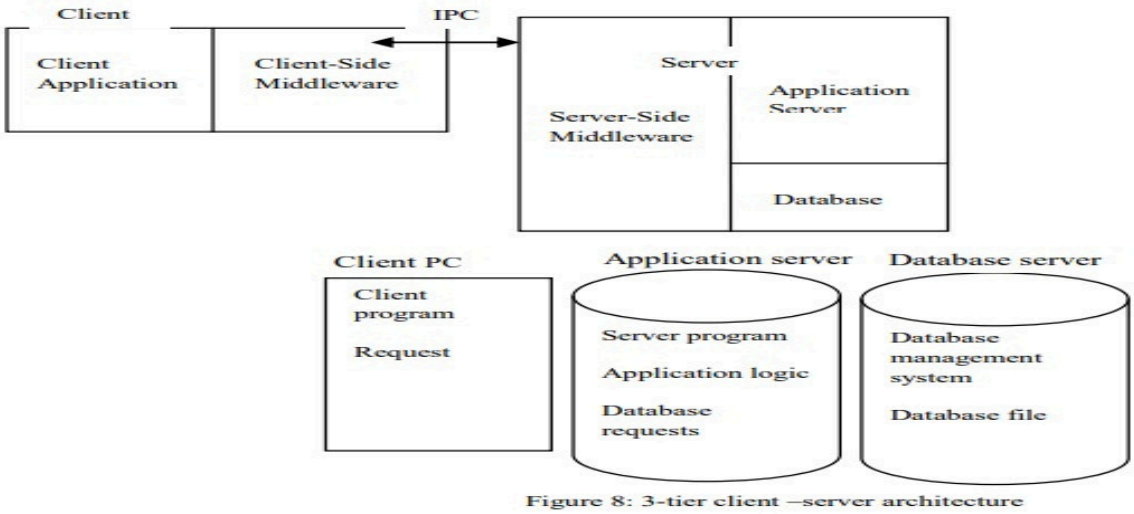


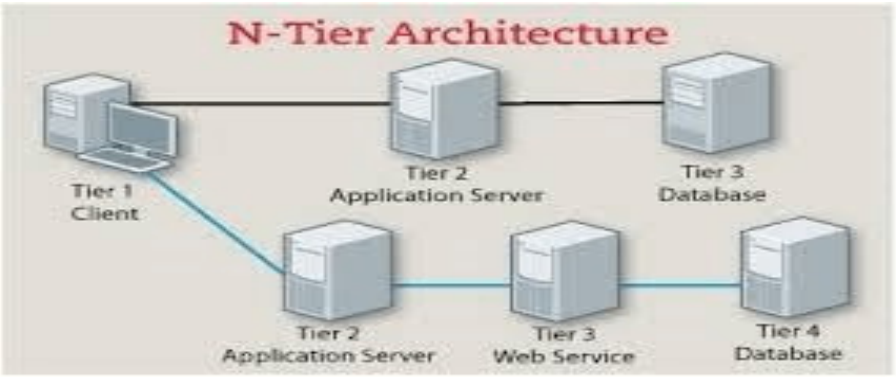
Figure 8: 3-tier client –server architecture

N-Tier Architecture

Description: An extension of the 3-Tier model, where the application is divided into multiple logical tiers (layers).

Characteristics: Further improves modularity, scalability, and maintainability by assigning specific functions to each layer. Physical tiers are also used, where each layer runs on a different physical machine.

Example: Large, complex enterprise systems and distributed applications.



.Recent Development

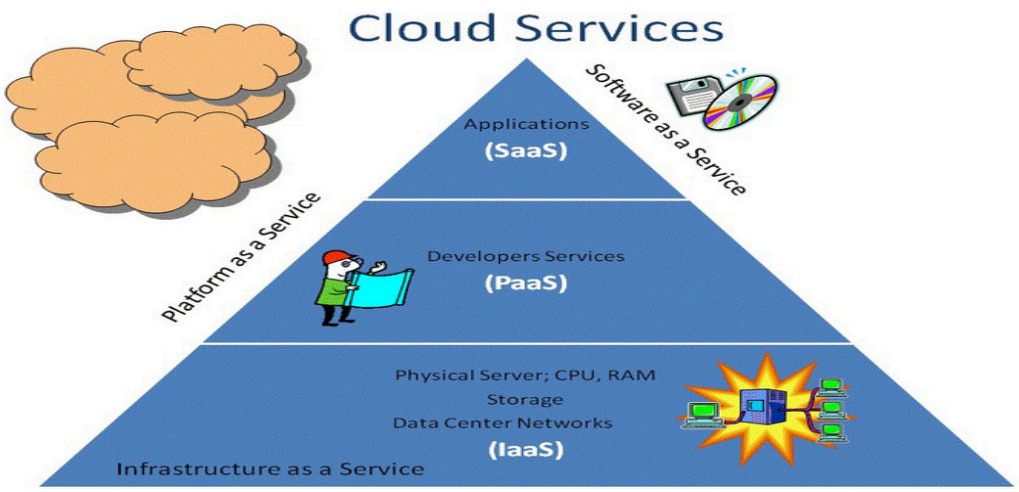
Client server continues to gain more mainstream adoption as more companies move into the cloud. Market growth in different types of service such as distributed and cloud computing.

Cloud computing: There are many definitions of cloud computing, but one of the most common ones describe cloud computing as a group of distributed computer that provides services and resources through the internet . There are three main services that can be offered by cloud, which are:

-Infrastructure as Service (IaaS): The products offered by this mode are through the Internet, such as servers.

- Platform as a Service (PaaS): In the platform of cloud computing, the services are provided through the Internet by cloud providers. Unlike, the traditional method in which each application requires the use of hardware, software, operating system etc.
- Software as a Service (SaaS): in this layer, you don't need to install or maintain software; the applications are delivered through the Internet.

The general importance of Client Server system in Cloud computing is horizontal scalability to millions of virtual machines, examples of applications that uses this technology include; Google Apps (Gmail, Google talk) . For example, in a university setting which has a computer centre that caters for the student, lecturers, software developer and researchers by providing them with the required software, hardware and development tools. The university decide to shift to the use of cloud, such that the students and lecturers can use the service of SaaS and IaaS clouds provider and then the software used by them will be saved on the servers of SaaS cloud and can be accessed online, then any other hardware or additional requirement will be executed online by the IaaS cloud provider. The software developer can finally use all the software and hardware they need for the development and hosting through PaaS cloud provider.



Mobile agent: This is an entity working with the computer, it has the ability to reason, can run in another remote site with the support of network, gather results, search results, work with other sites and finally return to its own site after the completion of its assigned tasks . They are called mobile agents because they have the ability to move from one computer to another computer through the network. Mobile agents serve as a direct extension of the client server approach. In the client-server approach, each communication entities have a specific role to play like the server offering services and the client using them. Mobile agents is a form of advancement to client-server system and it has more benefit, some of which are:

- Communication: In client-server system, the servers do not have the ability to communicate with each other, whereas mobile agent works like a peer to peer entity and can either act as a client or server.

- Persistence: When the mobile agent is created, it has the ability to work on its own so it is not affected when other nodes fails.

- Efficiency: This will reduce the traffic caused in client-server system during the process of sending messages, because the mobile agent has the ability to pre-process data locally and choose the important information to send.

- Fault tolerance: In the case of client-server system, when a server is down the connection is lost. But the mobile agents have the ability to continue working in the node if network fails.

Issues And Challenges In Client-Server System

Key issues and challenges in client-server systems include server dependency (a single server failure can halt the system), scalability constraints (servers can be overloaded), security vulnerabilities (data in transit or on the server can be compromised), network issues like congestion and latency that slow performance, high maintenance and hardware costs, and complexities in managing data integrity, concurrency, and client-server communication.

Server Dependency & Availability

Single Point of Failure: If the central server fails or goes down, all connected clients lose access to services and data, making the entire system unavailable.

Server Overload:

High volumes of simultaneous client requests can overwhelm the server, leading to slow performance, errors, and an inability to respond to new requests.

Scalability & Performance

Limited Scalability:

As the number of users or the complexity of requests increases, the server's capacity can become a bottleneck, making it difficult to scale the system effectively.

Network Congestion & Latency:

High network traffic or poor connection quality between clients and the server can cause delays, slow down data transmission, and negatively impact user experience.

Security

Vulnerability to Attacks:

Client-server systems are susceptible to cybersecurity threats, including DDoS attacks on the server, data interception during transmission, and unauthorized access.

Data Integrity:
Ensuring data is not corrupted or tampered with during transmission between the client and server requires robust encryption and error-checking mechanisms.

Costs & Maintenance
High Initial & Maintenance Costs:
The initial investment for powerful servers and ongoing costs for their maintenance, upgrades, and skilled IT personnel can be significant.

Hardware and Infrastructure:
The need to upgrade hardware (CPU, RAM, storage) as processing demands grow adds to the costs and complexity of managing the infrastructure.
Data Management

Concurrency Issues:
Managing multiple simultaneous requests to access or modify the same data can lead to conflicts and inconsistencies, requiring careful synchronization.

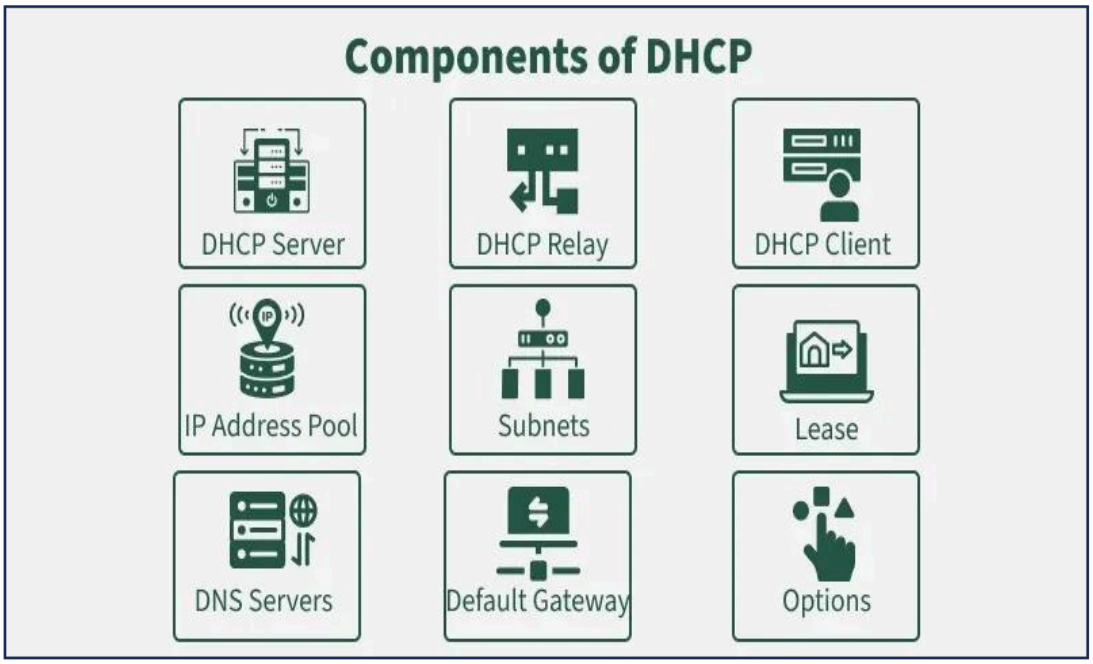
Data Access and Synchronization:
Ensuring all clients can consistently and reliably access and update shared data, especially in distributed environments, presents a significant challenge.

Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol is a network protocol used to automate the process of assigning IP addresses and other network configuration parameters to devices (such as computers, smartphones and printers) on a network.
Instead of manually configuring each device with an IP address, DHCP allows devices to connect to a network and receive all necessary network information, like IP address, subnet mask, default gateway and DNS server addresses, automatically from a DHCP server.

Components of DHCP

Themain components of DHCP include:



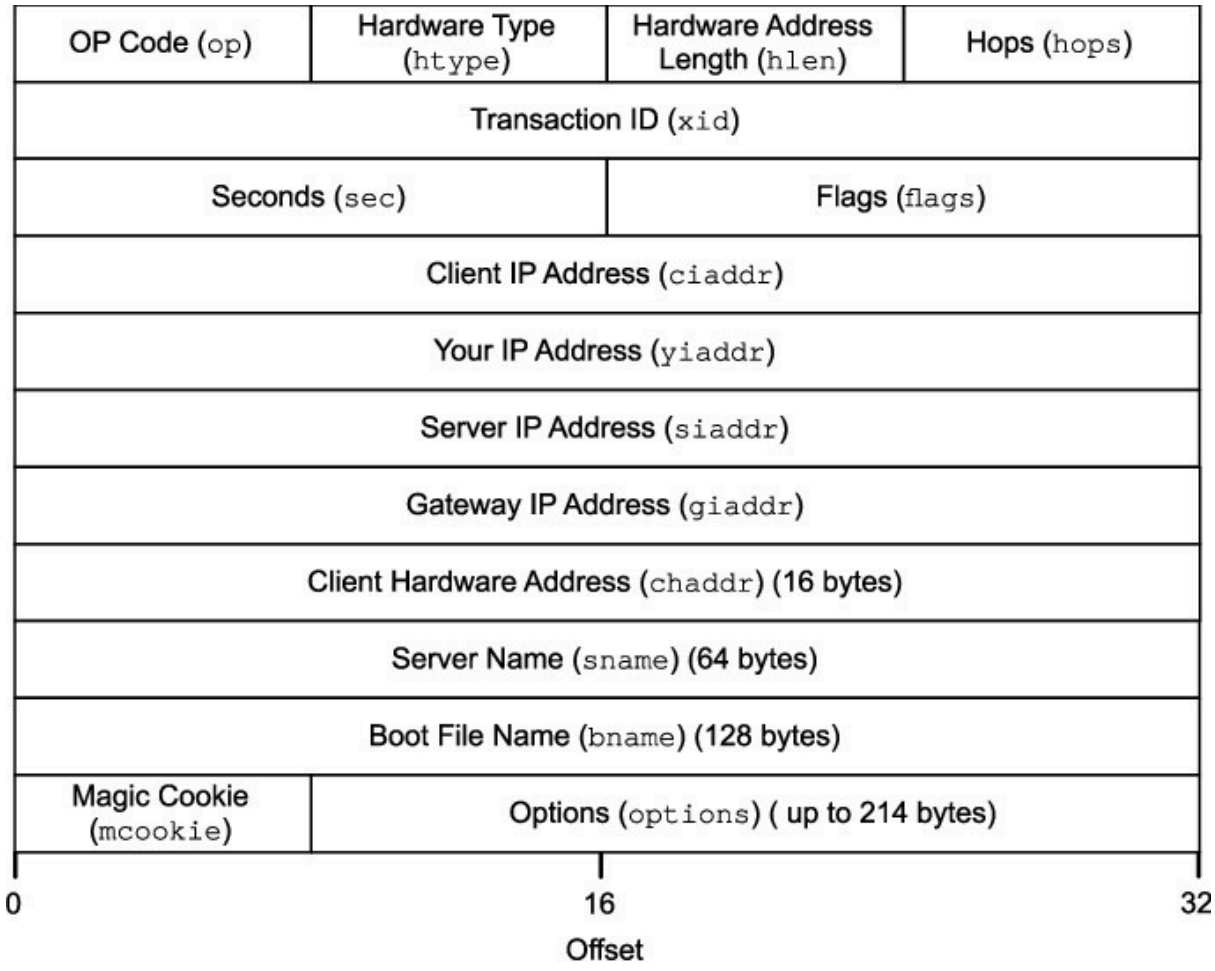
- DHCP Server: DHCP Server is a server that holds IP Addresses and other information related to configuration.
- DHCP Client: It is a device that receives configuration information from the server. It can be a mobile, laptop, computer or any other electronic device that requires a connection.

- DHCP Relay: DHCP relays basically work as a communication channel between DHCP Client and Server.
- IP Address Pool: It is the pool or container of IP Addresses possessed by the DHCP Server. It has a range of addresses that can be allocated to devices.
- Subnets: Subnets are smaller portions of the IP network partitioned to keep networks under control.
- Lease: It is simply the time that how long the information received from the server is valid, in case of expiration of the lease, the tenant must have to re-assign the lease.
- DNS Servers: DHCP servers can also provide DNS (Domain Name System) server information to DHCP clients, allowing them to resolve domain names to IP addresses.
- Default Gateway: DHCP servers can also provide information about the default gateway, which is the device that packets are sent to when the destination is outside the local network.
- Options: DHCP servers can provide additional configuration options to clients, such as the subnet mask, domain name and time server information.
- Renewal: DHCP clients can request to renew their lease before it expires to ensure that they continue to have a valid IP address and configuration information.
- Failover: DHCP servers can be configured for failover, where two servers work together to provide redundancy and ensure that clients can always obtain an IP address and configuration information, even if one server goes down.
- Dynamic Updates: DHCP servers can also be configured to dynamically update DNS records with the IP address of DHCP clients, allowing for easier management of network resources.
- Audit Logging: DHCP servers can keep audit logs of all DHCP transactions, providing administrators with visibility into which devices

are using which IP addresses and when leases are being assigned or renewed.

DHCP Packet Format.

Each field in the DHCP packet format plays a important role in enabling dynamic IP configuration, ensuring proper identification, addressing, timing, and communication between the client and server during the DHCP handshake process.

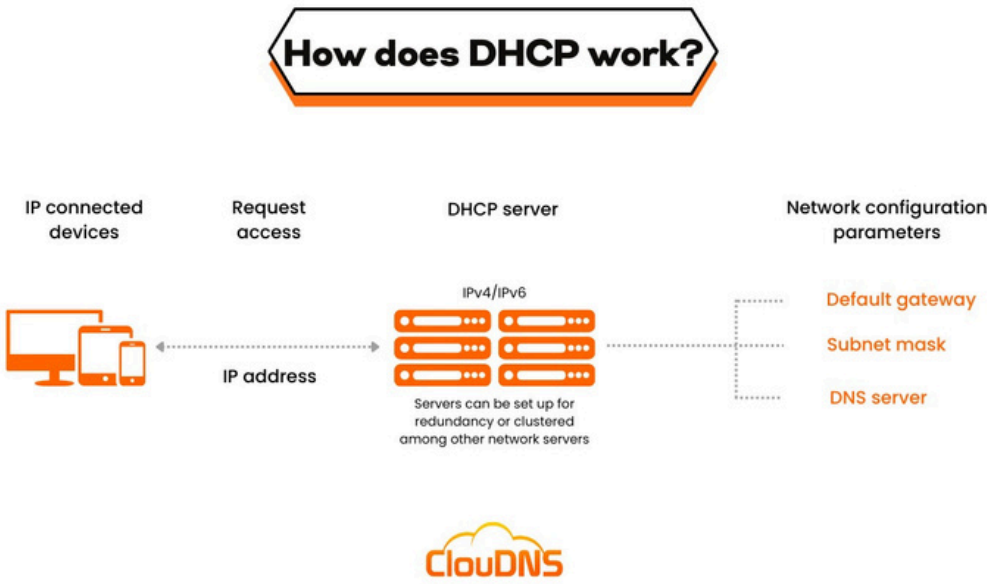


- Hardware Length: This is an 8-bit field defining the length of the physical address in bytes. e.g. for Ethernet the value is 6.
- Hop count: This is an 8-bit field defining the maximum number of hops the packet can travel.

- Transaction ID: This is a 4-byte field carrying an integer. The transaction identification is set by the client and is used to match a reply with the request. The server returns the same value in its reply.
- Number of Seconds: This is a 16-bit field that indicates the number of seconds elapsed since the time the client started to boot.
- Flag: This is a 16-bit field in which only the leftmost bit is used and the rest of the bit should be set to 0. A leftmost bit specifies a forced broadcast reply from the server.
- Client IP Address: This is a 4-byte field that contains the client IP address . If the client does not have this information this field has a value of 0.
- Your IP Address: This is a 4-byte field that contains the client IP address. It is filled by the server at the request of the client.
- Server IP Address: This is a 4-byte field containing the server IP address. It is filled by the server in a reply message.
- Gateway IP Address: This is a 4-byte field containing the IP address of a routers. IT is filled by the server in a reply message.
- Client Hardware Address: This is the physical address of the client .Although the server can retrieve this address from the frame sent by the client it is more efficient if the address is supplied explicitly by the client in the request message.
- Server Name: This is a 64-byte field that is optionally filled by the server in a reply packet. It contains a null-terminated string consisting of the domain name of the server. If the server does not want to fill this filed with data, the server must fill it with all 0s.
- Boot Filename: This is a 128-byte field that can be optionally filled by the server in a reply packet. It contains a null- terminated string consisting of the full pathname of the boot file.
- Options: This is a 64-byte field with a dual purpose. IT can carry either additional information or some specific vendor information. The field is used only in a reply message.

Working of DHCP

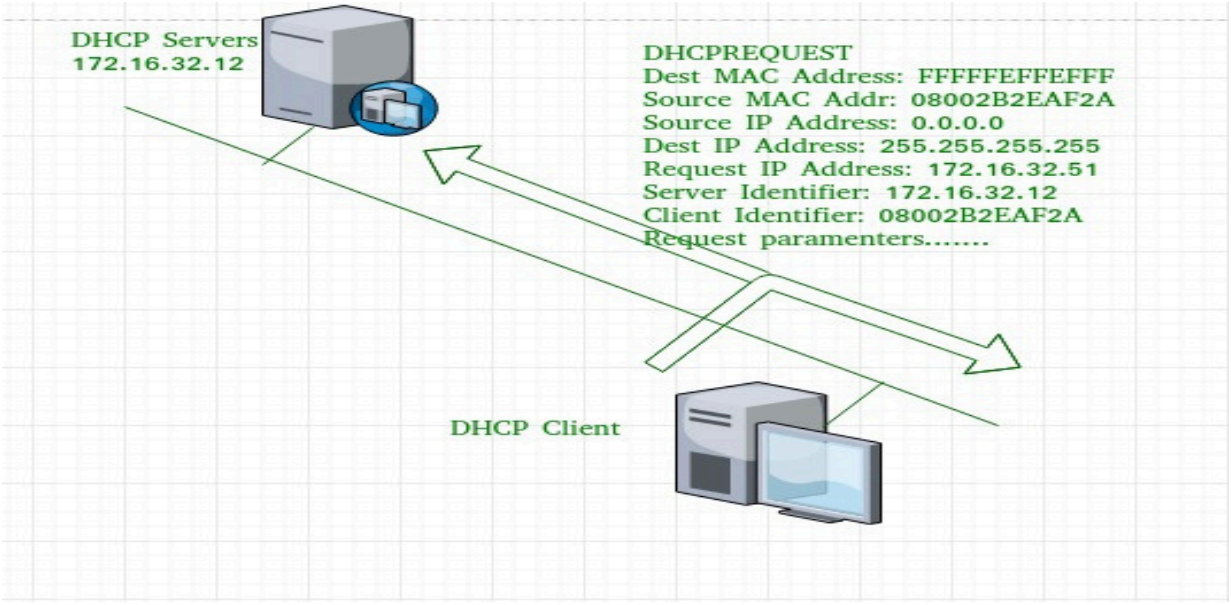
DHCP works on the Application layer of the UDP Protocol. The main task of DHCP is to dynamically assigns IP Addresses to the Clients and allocate information on TCP/IP configuration to Clients.



The DHCP port number for the server is 67 and for the client is 68. It is a client-server protocol that uses UDP services. An IP address is assigned from a pool of addresses. In DHCP, the client and the server exchange mainly 4 DHCP messages in order to make a connection, also called the DORA process, but there are 8 DHCP messages in the process.

The 8 DHCP Messages

1. DHCP Discover Message: This is the first message generated in the communication process between the server and the client. This message is generated by the Client host in order to discover if there is any DHCP server/servers are present in a network or not. This message is broadcasted to all devices present in a network to find the DHCP server. This message is 342 or 576 bytes long.



As shown in the figure, the source MAC address (client PC) is 08002B2EAF2A, the destination MAC address(server) is FFFFFFFFFFFF, the source IP address is 0.0.0.0(because the PC has had no IP address till now) and the destination IP address is 255.255.255.255 (IP address used for broadcasting). As they discover message is broadcast to find out the DHCP server or servers in the network therefore broadcast IP address and MAC address is used.

2. DHCP Offers A Message: The server will respond to the host in this message specifying the unleased IP address and other TCP configuration information. This message is broadcasted by the server. The size of the message is 342 bytes. If there is more than one DHCP server present in the network then the client host will accept the first DHCP OFFER message it receives. Also, a server ID is specified in the packet in order to identify the server.

Now, for the offer message, the source IP address is 172.16.32.12 (server's IP address in the example), the destination IP address is 255.255.255.255 (broadcast IP address), the source MAC address is 00AA00123456, the destination MAC address is 00:11:22:33:44:55 (client's MAC address). Here, the offer message is broadcast by the DHCP server therefore destination IP address is the broadcast IP address and destination MAC address is 00:11:22:33:44:55 (client's MAC address)and the source IP address is the server IP address and the MAC address is the server MAC address.

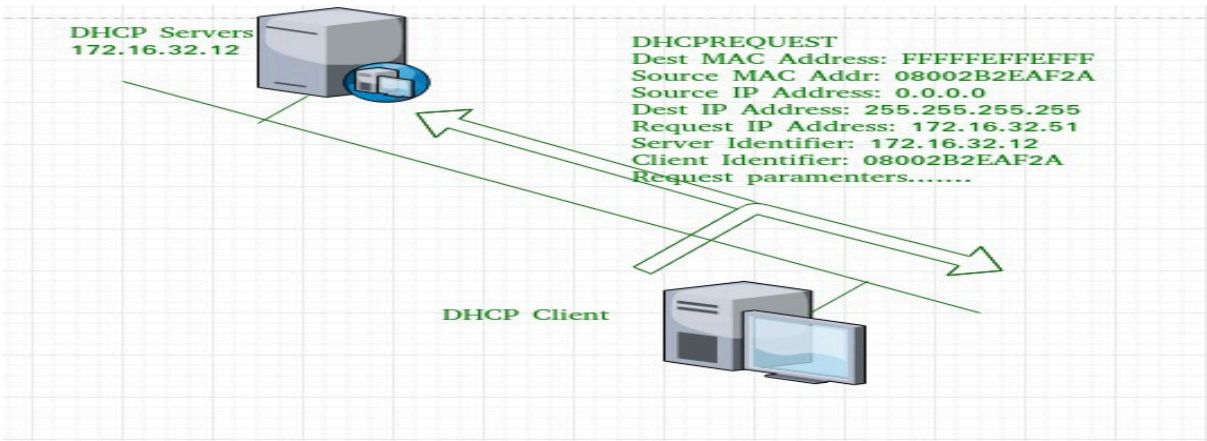
Also, the server has provided the offered IP address 192.16.32.51 and a lease time of 72 hours(after this time the entry of the host will be erased from the server automatically). Also, the client identifier is the PC MAC address (08002B2EAF2A) for all the messages.

3. DHCP Request Message: When a client receives an offer message, it responds by broadcasting a DHCP request message. The client will produce a gratuitous ARP in order to find if there is any other host present in the network with the same IP address. If there is no reply from another host, then there is no host with the same TCP configuration in the network and the message is broadcasted to the server showing the acceptance of the IP address. A Client ID is also added to th

Now, the request message is broadcast by the client PC therefore source IP address is 0.0.0.0(as the client has no IP right now) and destination IP address is 255.255.255.255 (the broadcast IP address) and the source MAC address is 08002B2EAF2A (PC MAC address) and destination MAC address is FFFFFFFFFFFF.

Note - This message is broadcast after the ARP request broadcast by the PC to find out whether any other host is not using that offered IP. If there is no reply, then the client host broadcast the DHCP request message for the server showing the acceptance of the IP address and Other TCP/IP Configuration.

4. DHCP Acknowledgment Message: In response to the request message received, the server will make an entry with a specified client ID and bind the IP address offered with lease time. Now, the client will have the IP address provided by the server.



Now the server will make an entry of the client host with the offered IP address and lease time. This IP address will not be provided by the server to any other host. The destination MAC address is 00:11:22:33:44:55 (client's MAC address) and the destination IP address is 255.255.255.255 and the source IP address is 172.16.32.12 and the source MAC address is 00AA00123456 (server MAC address).

5. DHCP Negative Acknowledgment Message: Whenever a DHCP server receives a request for an IP address that is invalid according to the scopes that are configured, it sends a DHCP NACK message to the client. E.g. when the server has no IP address unused or the pool is empty, then this message is sent by the server to the client.

6. DHCP Decline: If the DHCP client determines the offered configuration parameters are different or invalid, it sends a DHCP decline message to the server. When there is a reply to the gratuitous ARP by any host to the client, the client sends a DHCP decline message to the server showing the offered IP address is already in use.

7. DHCP Release: A DHCP client sends a DHCP release packet to the server to release the IP address and cancel any remaining lease time.

8. DHCP Inform: If a client address has obtained an IP address manually then the client uses DHCP information to obtain other local configuration parameters, such as domain name. In reply to the DHCP inform message, the DHCP server generates a DHCP ack message with a local configuration suitable for the client without allocating a new IP address. This DHCP ack message is unicast to the client.

Security Measures for Using DHCP

To make sure your DHCP servers are safe, consider these DHCP security issues:

Limited IP Addresses : A DHCP server can only offer a set number of IP addresses. This means attackers could flood the server with requests, causing essential devices to lose their connection.

Fake DHCP Servers : Attackers might set up fake DHCP servers to give out fake IP addresses to devices on your network.

DNS Access : When users get an IP address from DHCP, they also get DNS server details. This could potentially allow them to access more data than they should.

Security Measures for Using DHCP

To make sure your DHCP servers are safe, consider these DHCP security issues:

Limited IP Addresses : A DHCP server can only offer a set number of IP addresses. This means attackers could flood the server with requests, causing essential devices to lose their connection.

Fake DHCP Servers : Attackers might set up fake DHCP servers to give out fake IP addresses to devices on your network.

DNS Access : When users get an IP address from DHCP, they also get DNS server details. This could potentially allow them to access more data than they should

Domain Name System (DNS)

DNS, or Domain Name System, is a hierarchical and distributed naming system that translates human-friendly domain names (like www.example.com) into machine-readable IP addresses (like 192.0.2.1). This system allows users to access websites and other resources on the internet using easy-to-remember names instead of numerical IP addresses.

Why is DNS important?

Internet operations depend on DNS, which functions like a virtual phone book and is the mapping apparatus that enables users and resources to identify and connect with each other. Internet traffic flows are highly dependent on DNS behind-the-scenes mapping that can automatically connect and direct internet traffic from point to point. Without DNS automatic mapping facilities, internet users and service requesters would have to know and manually enter the IP addresses of the services and sites to which they wish to connect. Additionally, DNS servers apply security verifications that can prevent security attacks and attempts.

How DNS works

DNS servers convert URLs and domain names into IP addresses that computers can understand and use. They translate what a user types into a browser into something the machine can use to find a webpage. This process of translation and lookup is called DNS resolution. Its purpose is to translate a human language-based name, like techtarget.com, into a numerical IP address that TCP/IP requires to locate a website or other internet resource.

There are two types of DNS resolution techniques:

1. Recursive resolution. In DNS recursive resolution, the human-readable domain name is translated into a numbered IP address by recursive resolvers. The user workstation, device or IT service sends the human-readable domain name identifier of the requested internet resource to the local network DNS server, which consults its database to find the associated IP address for that identifier. The DNS server uses the IP address to go out to the internet and retrieve the requested website or resource.

2. Iterative resolution. In the case of iterative resolution, the initial request that the user workstation, device or IT service sends to its local network DNS server is unsuccessful, likely because the internet resource being requested is on a different network that is governed by its own DNS server. In this case, the initial request is iteratively sent to several different DNS servers on different networks until the DNS server that contains the cross-reference and IP address translation for the requested internet resource is found. The IP address is then used to retrieve the requested resource for the requester.

The basic process of DNS resolution follows these steps:

1. The user enters a web address or domain name into a browser.
2. The browser sends a recursive DNS query message to the network to determine to which IP address or network the domain corresponds.
3. The query goes to a recursive DNS server, which is usually managed by the ISP. If the recursive resolver has the address, it returns it to the user, and the webpage loads.

4. If the recursive DNS server has no answer, it queries a series of other servers in the following order: DNS root name servers, TLD name servers and authoritative name servers.

5. The three server types work together and continue redirecting until they retrieve a DNS record that contains the queried IP address. They send this information to the recursive DNS server, and the webpage the user is looking for loads. DNS root name servers and TLD servers primarily redirect queries and rarely provide the resolution themselves.

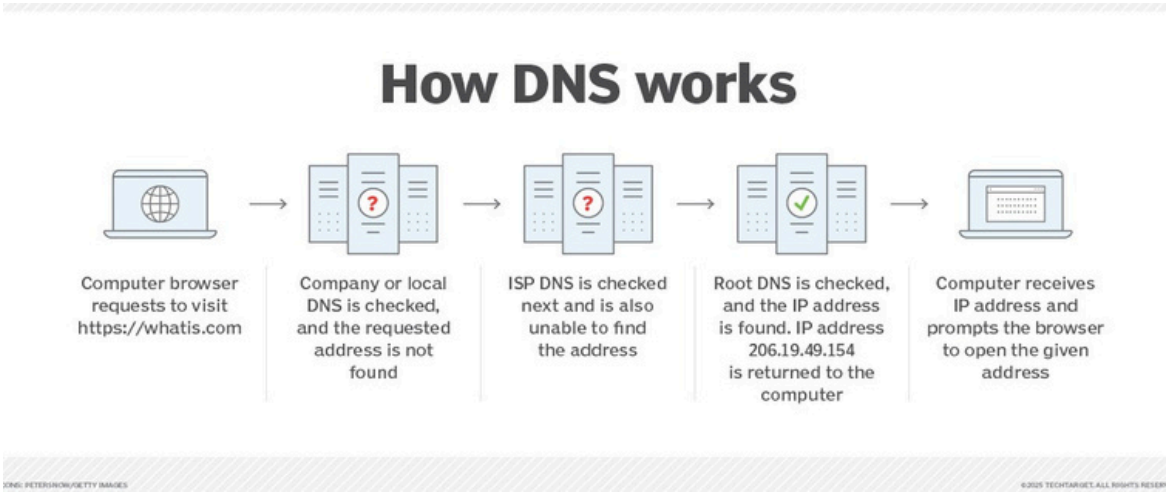
6. The recursive server stores, or caches, the IP address for the domain name. The next time it receives a request for that domain name, it can respond directly to the user instead of querying other servers.

7. If the query reaches the authoritative server and it cannot find the information, it returns an error message.

The entire process of querying the various servers takes a fraction of a second and is usually imperceptible to the user.

DNS servers answer questions from both inside and outside their domains. When a server receives a request from outside the domain for information about a name or address inside the domain, it provides an authoritative answer.

When a server gets a request from within its domain for a name or address outside that domain, it forwards the request to another server, usually one managed by its ISP.



DNS server types

DNS uses different servers to locate the IP addresses of the domain names that users request. The following is a breakdown of how these various DNS servers work together:

1. Recursive server. This DNS server is within the same network as the user, so it is the first DNS server that attempts translation of the domain name submitted by the user into an IP address. The user enters www.getthis.com. The request goes out to the recursive server, which searches its cross-reference database of domain names and IP addresses. Unfortunately, the recursive server that is on the user's network cannot find the IP address domain name www.getthis.com.

2. Root name server. The recursive server on the user's network then reaches out to the root name server, which is a master index of all the servers with the information being queried. The Internet Corporation for Assigned Names and Numbers, or ICANN, oversees these servers. The root server looks at the TLD of the resource being requested -- for example, .com, .org or .edu.

3. TLD server. Based on the TLD name of the resource requested, the root server calls the correct TLD server. For example, www.getthis.com has .com as its TLD name, so the root server routes the user's request to the TLD server that contains an IP number cross-reference database for all .com domains.

4. Authoritative server. The authoritative server is the final authority for all internet assets, as it holds the DNS records for the sites and resources that users access. The authoritative server works with recursive servers, root servers and TLD servers to return the full resource or website requested by the user. In the www.getthis.com example, the resource request was first routed to a recursive server, which could not find the website. It was then forwarded to the root server, which contained a master index of DNS names, and then directed the request to a .com domain TLD server. The .com TLD server found the www.getthis.com domain name and its corresponding IP address and then contacted the authoritative server, which contained the domain

itself. It was able to fulfill the user request by facilitating the complete delivery of the requested domain content to the user.

Types of DNS queries

Recursive and iterative queries are the two types of queries most often executed when users or IT services request internet resources from DNS. However, the following other types of DNS queries can also occur:

Non recursive queries. These are queries for which the recursive resolver server already knows where to get the answer. The answer is either cached on the recursive server itself, or the recursive server knows to skip the root and TLD servers and go directly to a specific authoritative server. The query is non recursive because there is no need -- and, therefore, no request -- for any more queries. Non recursive queries resolve in the answer. If a recursive resolver has cached an IP address from a previous session and serves that address upon the next request, that is considered a nonrecursive query.

Records not found queries. These are cases where the recursive, root level, TLD and authoritative servers have worked together to locate a website or resource requested, but they cannot find it. It is possible that a user might have mistyped a domain name, or the domain name requested does not exist. In these cases, a not found error message is returned to the user.

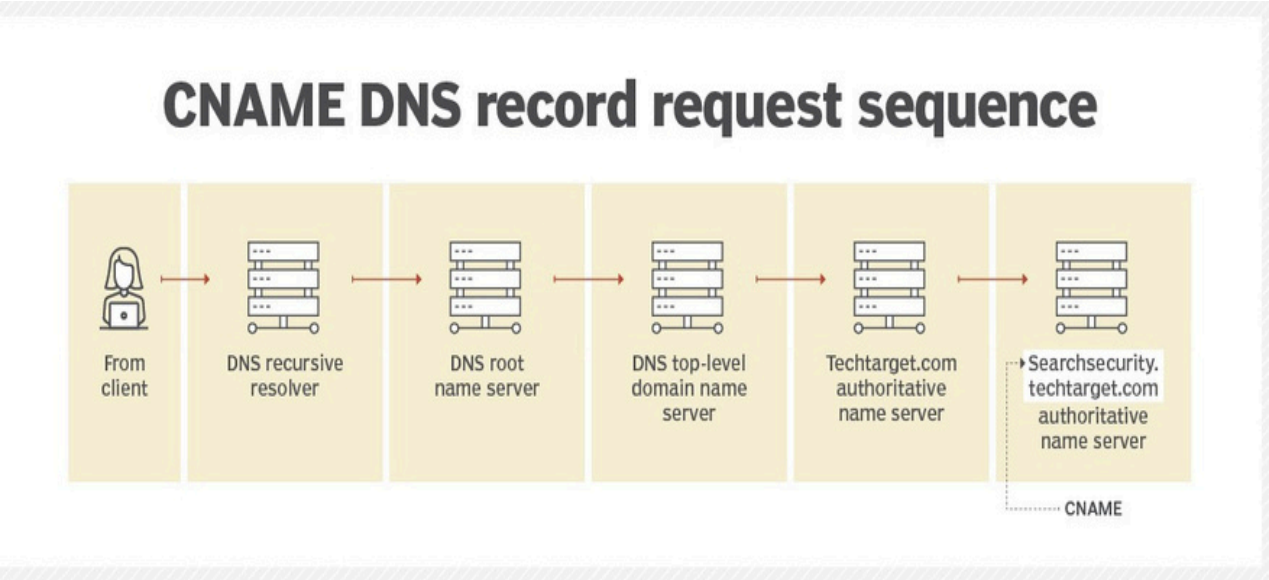
DNS not responding queries. These queries occur when DNS servers are either down or cannot be reached. Sometimes, this is a systemic problem that users or IT cannot directly fix, but other times, it is possible to get DNS working again by either trying to access the internet resource requested through an alternate web browser or by rebooting a local router.

Common DNS records

DNS records are the information a query seeks. Depending on the query, client or application, different information is required. There are many DNS record types, each with its own purpose in denoting how a query should be treated. The following are common DNS records:

- A or AAAA record. This stands for address and holds the IP address of a domain. These records only apply to IP addresses that are registered on IP version 4 (IPv4), which most companies are still running. The issue with IPv4 is that it has a finite number of IP addresses that can run on it, but its eventual replacement, IPv6, has the potential to carry limitless IP addresses with stronger security. On IPv6, the address record known as the A record on IPv4 is called the AAAA record and uses the longer IP address format of IPv6 addresses. Most websites only have one A or AAAA record, but some larger sites have several. This helps with load balancing because different A or AAAA records can be used for different users in heavy traffic.
- Name server record. The DNS name server record denotes which authoritative DNS server is responsible for having all the information about a given domain so users can be routed to the website or resource they are requesting. Given the importance of the DNS authoritative name server, it is not uncommon for domains to use both primary and backup name servers to increase reliability. Multiple name server records are created to enable queries to be routed to different DNS authoritative name servers.
- TXT record. TXT records enable administrators to enter text into DNS. The original purpose was to put human-readable notes in DNS, but today, machine-readable notes are often put there. TXT records are used to confirm domain ownership, secure email and prevent email spam.
- Canonical name record. CNAME records are used to resolve situations where there might be multiple domain names or aliases for a particular website or internet resource. A CNAME search expands the DNS search to alternate domain names besides the original name the user entered or what was loaded into the A or AAAA record. By using the CNAME records to try out alternative alias names for a requested website or resource, DNS servers enhance their chances of locating the correct name of the website or resource originally requested. An example is if a user keyed in a URL of searchsecurity.techtarget.com. In the event the DNS server could not locate this name and its corresponding IP address

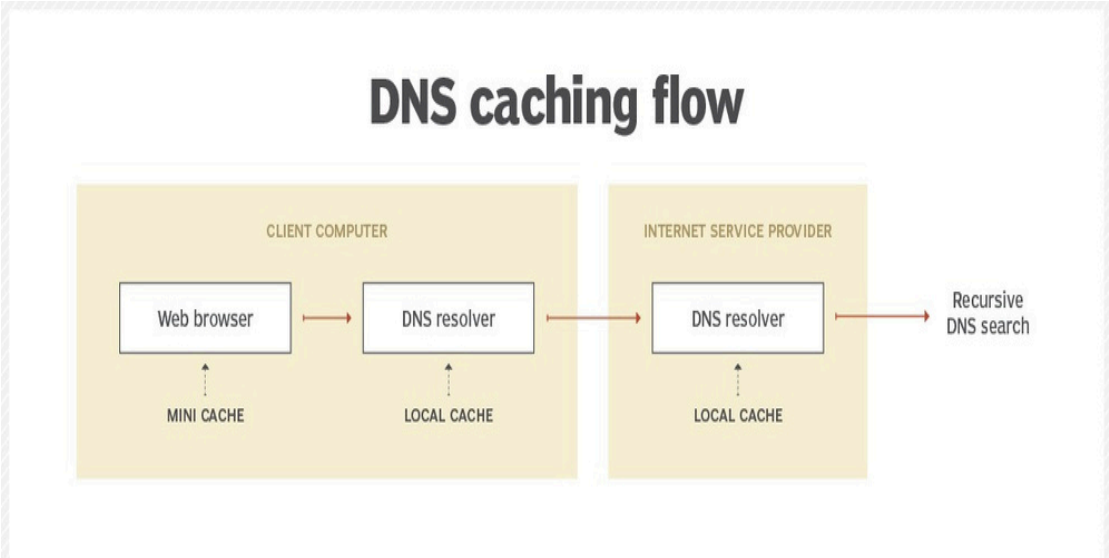
for routing, the server queries the CNAME aliases, such as techtarget.com.



How does DNS increase web performance? Servers can cache the A or AAAA records or IP addresses they receive from DNS queries for a set time. Caching promotes efficiency, enabling servers to respond quickly the next time a request for the same IP address comes in. For example, if everyone in an office needs to access the same training video on a particular website on the same day, the local DNS server only has to resolve the name once, and then it can serve all the other requests out of its cache. The length of time the record is held -- also known as the time to live (TTL) -- is set by administrators and depends on various factors. Longer time periods decrease the load on servers, and shorter ones ensure the most accurate responses

DNS caching DNS caching aims to reduce the time it takes to get an answer to a DNS query.
Caching enables DNS to store previous answers to queries closer to clients and get that information to them faster the next time it is queried. DNS data can be cached in the following places:

- Browser. Most browsers, like Apple Safari, Google Chrome and Mozilla Firefox, cache DNS data by default for a set amount of time. The browser is the first cache that gets checked when a DNS request is made -- before the request leaves the machine for a local DNS resolver server.
- Operating system. Many OSes have built-in DNS resolvers called stub resolvers that cache DNS data and handle queries before sending them to an external server. The OS is usually queried after the browser or other querying application.
- Recursive resolver. The answer to a DNS query can also be cached on the DNS recursive resolver. Resolvers might have some of the records necessary to return a response and be able to skip some steps in the DNS resolution process. For example, if the resolver has A or AAAA records but not the corresponding name server records, it can skip the root server and query the TLD server directly.



DNS security.

DNS security addresses protecting the Domain Name System (DNS) from various vulnerabilities and attacks that can compromise network integrity and data privacy. Key DNS security threats include:

DNS Spoofing and Cache Poisoning: Attackers insert false DNS data, redirecting users to malicious sites, risking data theft and malware infections.

DDoS Attacks: Overwhelming DNS servers with massive traffic to disrupt service availability.

DNS Tunaeling: Using DNS queries to covertly exfiltrate data or communicate with compromised devices.

DNS Hijacking: Redirecting DNS queries to malicious servers to intercept or alter traffic.

Domain Lockup: Exhausting server resources by continuous junk traffic.

Best Practices for DNS Security

Use DNSSEC: Adds cryptographic signatures to DNS data, ensuring authenticity and integrity.

Apply Access Controls: Restrict and monitor access to DNS servers, hiding sensitive information and limiting query sources.

Implement Redundancy: Deploy multiple DNS servers with failover to maintain availability

Regular Updates and Patching: Keep DNS software current to mitigate vulnerabilities.

Monitor and Log DNS Traffic: Detect suspicious activity early for rapid incident response.

Mitigate DDoS: Use traffic filtering and third-party services to absorb and block large-scale attacks.

Thesemeasures protect DNS infrastructure, maintain reliable name resolution, and safeguard user connections from cyber threats.

DHCP & DNS Setup

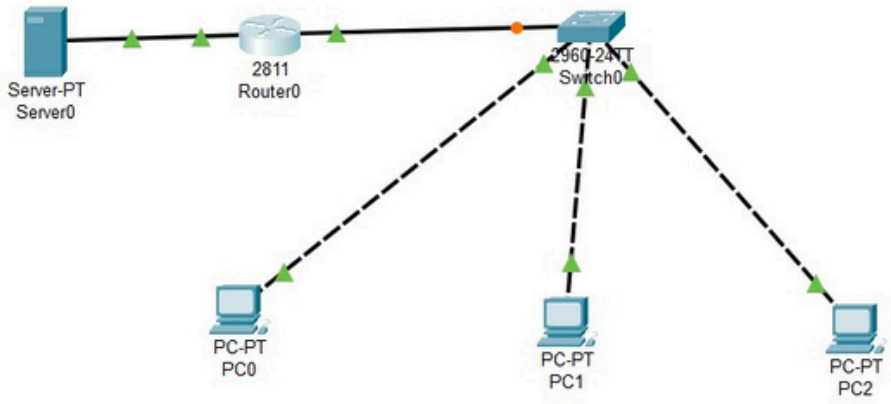
Step 1: Add Devicesand Connect

Open Cisco Packet Tracer.

Add 1 Router, 1 Switch, 1 Server, and 3 PCs from the devices list.

Connect all 3 PCs and the Server to the Switch using Copper Straight-Through cables.

Connect the Switch to the Router's FastEthernet0/0 using another Copper Straight-Through cable.

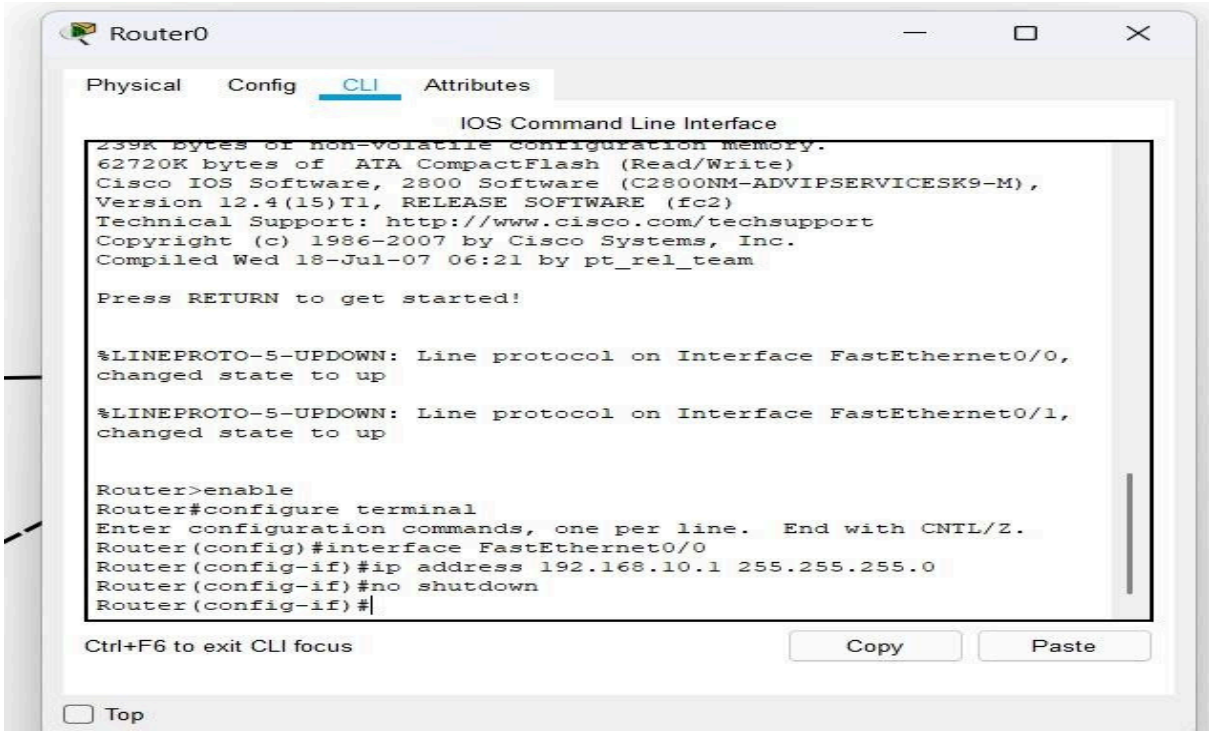


Step2. Configure the Router

Click on Router → CLI tab.

Enter the following commands:

```
>enable
>configure terminal
>interface FastEthernet0/0
>ip address 192.168.10.1 255.255.255.0
>no shutdown
>exit
```



Step3. Configure the Server

Set Static IP

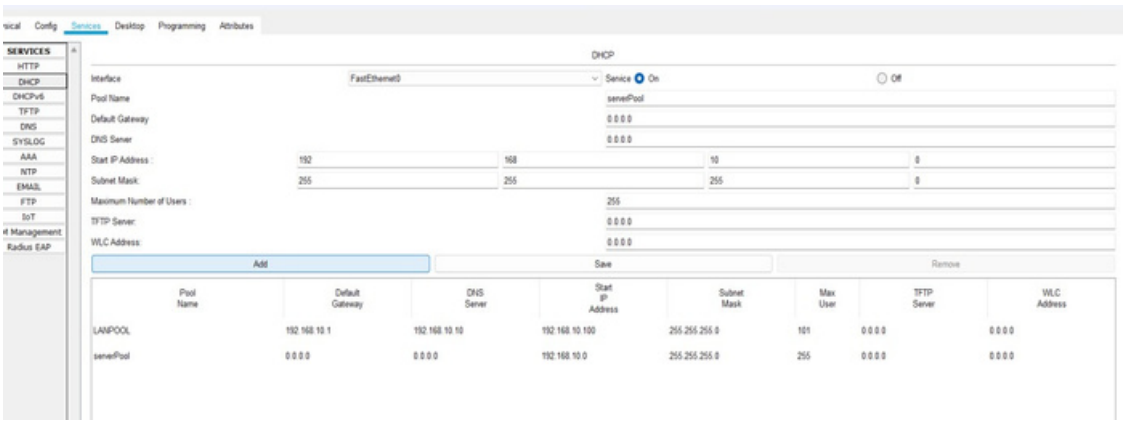
Click Server → Desktop → IP Configuration.

Set:

IP: 192.168.10.10

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.10.1.



Enable DHCP

Server → Services tab → DHCP.

Create Pool:

Pool Name: LANPOOL

Default Gateway: 192.168.10.1

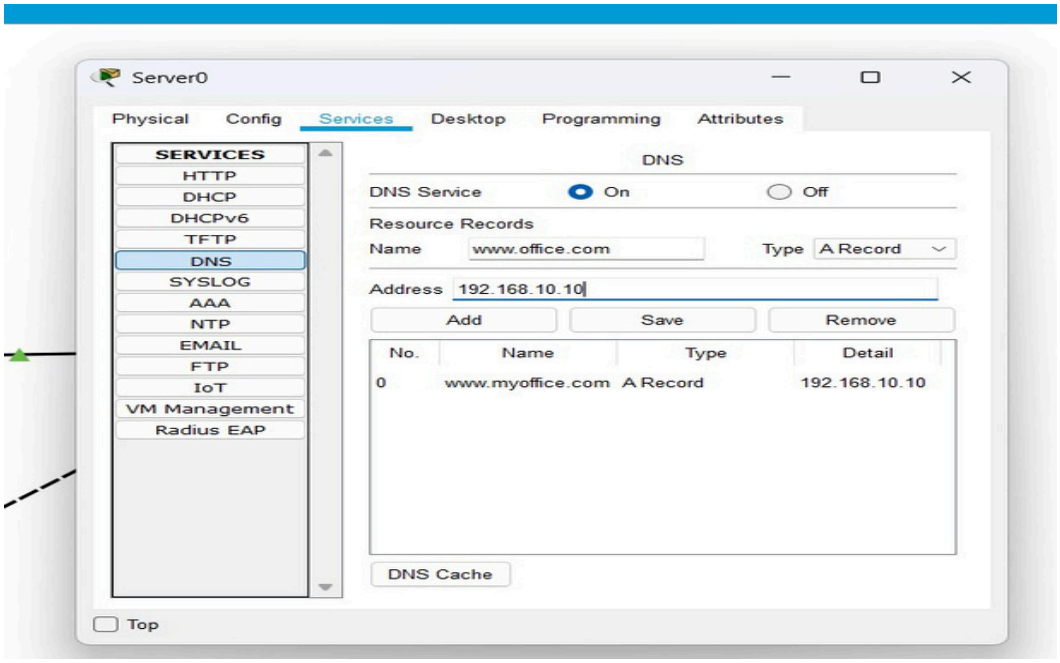
DNS Server: 192.168.10.10

Start IP Address: 192.168.10.100

Subnet Mask: 255.255.255.0

Maximum Users: (e.g., 101)

Click "Add" and "Save", then turn DHCP service ON

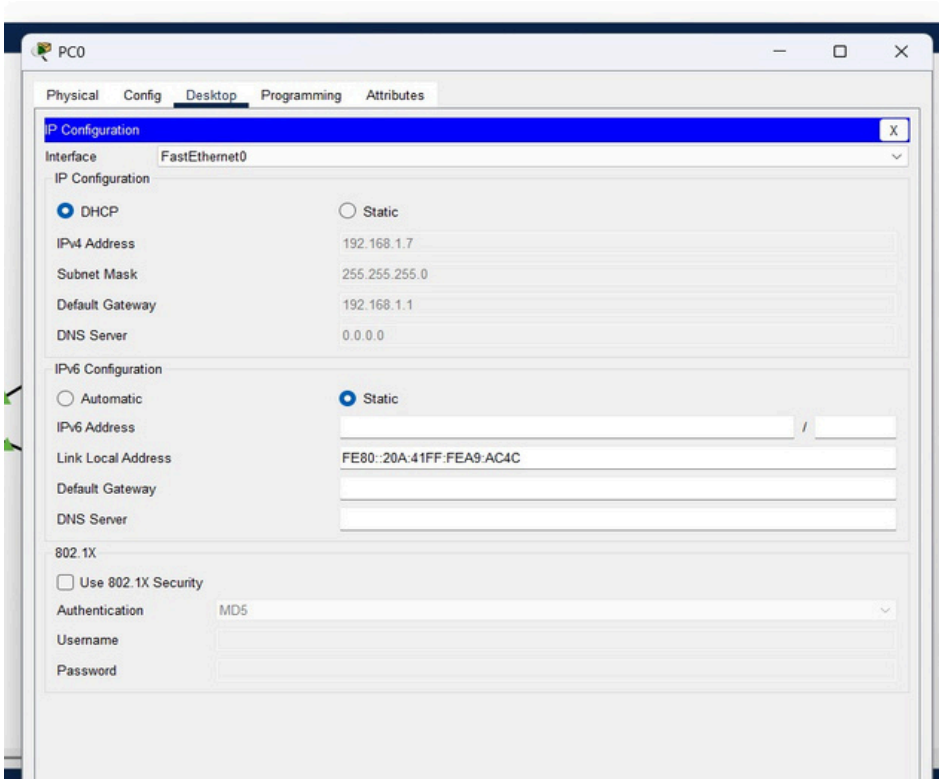


4. Configure the PCs for DHCP

For each of the three PCs, perform the following steps:

1. Click on the PC icon.
2. Go to the Desktop tab and select IP Configuration.
3. Select the DHCP option. The PC will send a DHCP request.

4. After a moment, the fields should automatically populate with an IP address from the 192.168.10.100-200 range, the correct subnet mask, and the gateway/DNS server addresses you configured.



5. Test the Network

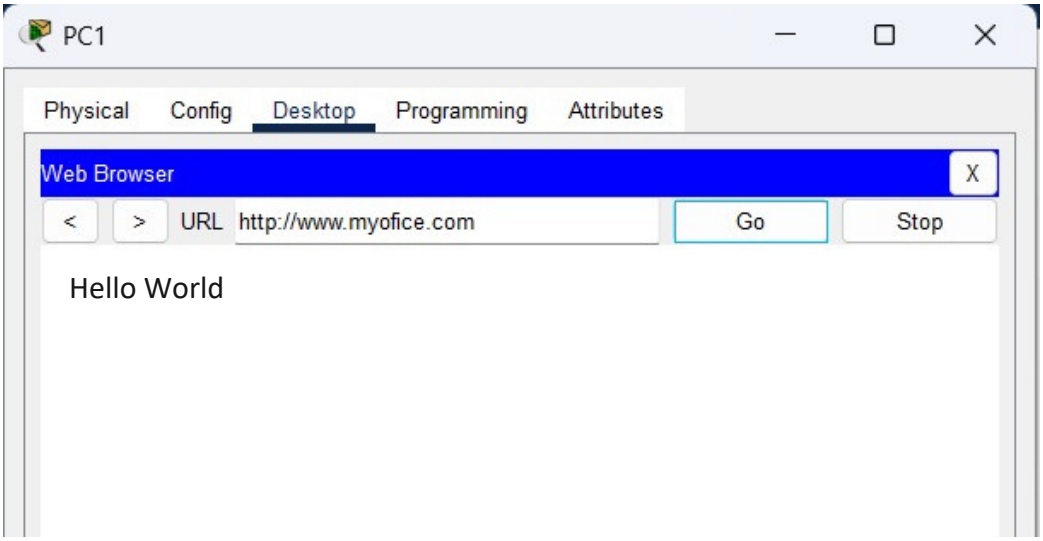
Your network should now be functional. You can verify this in two ways:

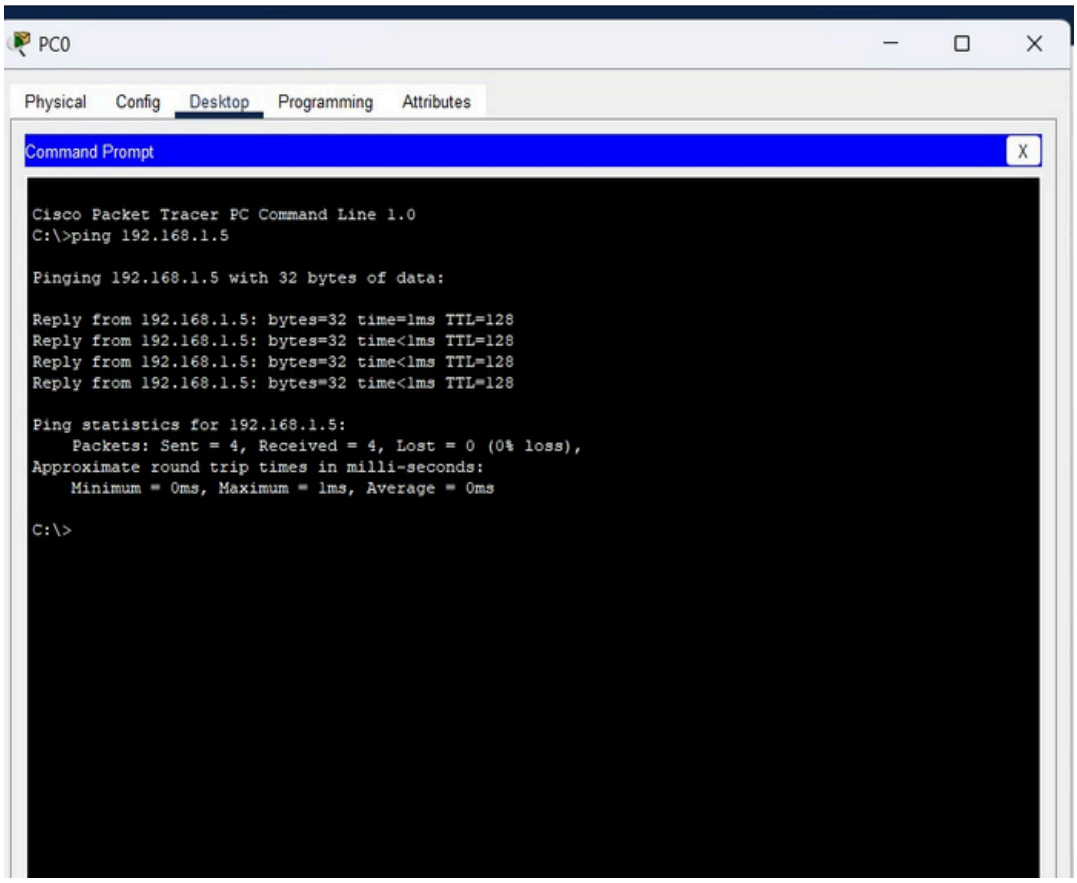
A. Test DHCP

Check each PC's IP configuration. They should have successfully received unique IP addresses like 192.168.10.100, 192.168.10.101, etc.

B. Test DNS and Connectivity

1. Click on any of the configured PCs.
2. Go to the Desktop tab and open the Web Browser.
3. In the URL bar, type www.myoffice.com and click Go.
4. The browser should display the default "Hello World" page from the server, confirming that DNS resolution and IP connectivity are working correctly. You can also test by pinging one PC from another using the Command Prompt tool.


PC1
Physical
Config
Desktop
Programming
Attributes
Web Browser
X
<
>
URL
http://www.myofice.com
Go
Stop
Hello World


PC0
Physical
Config
Desktop
Programming
Attributes
Command Prompt
X
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.5
Pinging 192.168.1.5 with 32 bytes of data:
Reply from 192.168.1.5: bytes=32 time=1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Reply from 192.168.1.5: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.5:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\>