Network Management and Administration
Code:3360703

UNIT – II

Network Protocols and Services

Prepared By:
Chirag Patel
Lecturer in Computer Engineering,
K D Polytechnic, Patan

Address Resolution Protocol(ARP)

- Mapping Logical to Physical Address
- Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network.
- This means that the sender needs the physical address of the receiver.
- The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network.
- Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet.
- The response packet contains the recipient's IP and physical addresses.
- The packet is unicast directly to the inquirer by using the physical address received in the query packet.

REVERSE ADDRESS RESOLUTION PROTOCOL (RARP)

- (RARP) finds the logical address for a machine that knows only its physical address.
- To create an IP datagram, a host or a router needs to know its own IP address or addresses.
- The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol.
- A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply.
- The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.

REVERSE ADDRESS RESOLUTION PROTOCOL (RARP)

- There is a serious problem with RARP: Broadcasting is done at the data link layer. The physical broadcast address, associates in the case of Ethernet, does not pass the boundaries of a network.
- This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet.
- This is the reason that RARP is almost obsolete
- Two protocols, BOOTP and DHCP, are replacing RARP

BOOTP (BOOTSTRAP PROTOCOL)

- The Bootstrap Protocol (BOOTP) is a client/server protocol designed to provide physical address to logical address mapping. BOOTP is an application layer protocol.
- The administrator may put the client and the server on the same network or on different networks
- One of the advantages of BOOTP over RARP is that the client and server are application–layer processes.
- As in other application–layer processes, a client can be in one network and the server in another, separated by several other networks. However, there is one problem that must be solved.
- The BOOTP request is broadcast because the client does not know the IP address of the server.
- A broadcast IP datagram cannot pass through any router.

BOOTP (BOOTSTRAP PROTOCOL)

- To solve the problem, there is a need for an intermediary. One of the hosts (or a router that can be configured to operate at the application layer) can be used as a relay.
- The host in this case is called a relay agent.
- The relay agent knows the unicast address of a BOOTP server. When it receives this type of packet, it encapsulates the message in a unicast datagram and sends the request to the BOOTP server. The packet, carrying a unicast destination address, is routed by any router and reaches the BOOTP server.
- The BOOTP server knows the message comes from a relay agent because one of the fields in the request message defines the IP address of the relay agent.
- The relay agent, after receiving the reply, sends it to the BOOTP client.

DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

- BOOTP is not a dynamic configuration protocol.
- When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address.
- However, what if a host moves from one physical network to another?
- What if a host wants a temporary IP address? BOOTP cannot handle these situations because the binding between the physical and IP addresses is static and fixed in a table until changed by the administrator. BOOTP is a static configuration protocol.
- The Dynamic Host Configuration Protocol (DHCP) has been devised to provide static and dynamic address allocation that can be manual or automatic.
- Static Address Allocation In this capacity DHCP acts as BOOTP does.

DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

- A DHCP server has a database that statically binds physical addresses to IP addresses.
- Dynamic Address Allocation DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.
- When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned.
- On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.

DHCP (DYNAMIC HOST CONFIGURATION PROTOCOL)

- DHCP provides temporary IP addresses for a limited time.
- The addresses assigned from the pool are temporary addresses. The DHCP server issues a lease for a specific time. When the lease expires, the client must either stop using the IP address or renew the lease.
- The server has the option to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

DHCP Objectives

- The DHCP server should be able to provide a workstation with all the settings needed to configure the TCP/IP client so that no manual configuration is needed.
- The DHCP server should assign IP addresses in such a way to prevent the duplication of addresses on the network.
- The DHCP server should be able to configure clients on other subnets through the use of relay agent.
- DHCP clients should be able to retain their TCP/IP parameters despite a reboot of either client or server system.

DHCP Packet Format

IP address assignment types in DHCP

- The Primary function of DHCP is to assign IP addresses and to accommodate the needs of all types of client system.
- There are three types of address assignment:
- **Manual allocation**
- The administrator configures the DHCP server to assign a specific IP address to a given system, which will never change unless it is manually modified. This is equivalent in functionality to RARP and BOOTP.
- **Automatic allocation**
- The DHCP server assigns permanent IP addresses from a pool, which does not change unless they are manually modified by the administrator.
- **Dynamic allocation**
- The DHCP server assigns IP addresses from a pool using a limited–time lease, so the addresses can be reassigned if the client system does periodically renew it.

DHCP Architecture

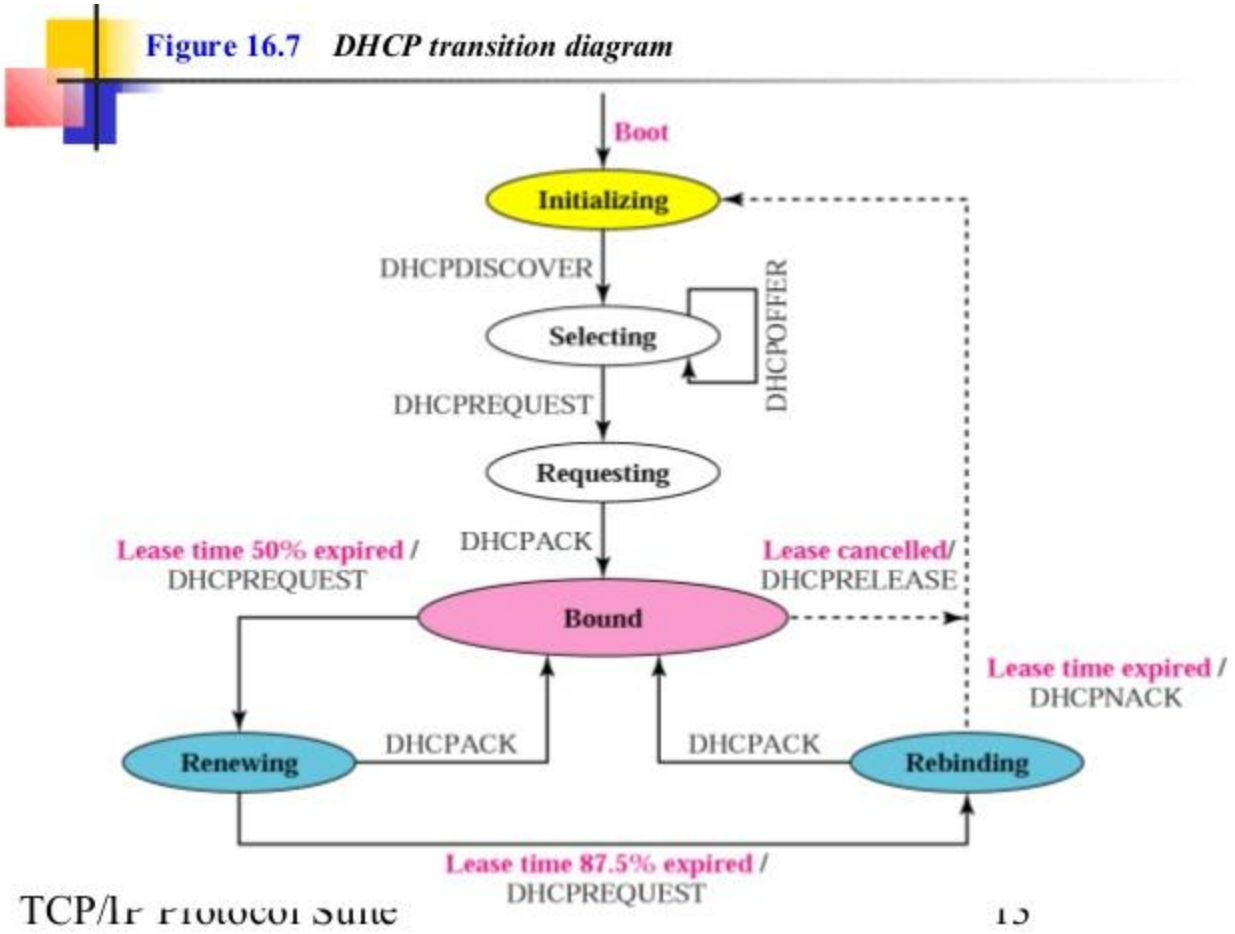
- The DHCP architecture consists of DHCP clients, DHCP servers, and DHCP relay agents on a network.
- **DHCP client functionality**
- The clients interact with servers using DHCP messages in a DHCP conversation to obtain and renew IP address leases.
- A DHCP client is any network-enabled device that supports the ability to communicate with a DHCP server in compliance with RFC 2131, for the purpose of obtaining dynamic leased IP configuration and related optional information.
- **DHCP server responsibilities**
- The DHCP servers maintain scopes, reservations, and options as set by the administrator.

DHCP Architecture

- **DHCP Relay Agent:**
- A relay agent is a small program that relays DHCP/BOOTP messages between clients and servers on different subnets.
- In TCP/IP networking, routers are used to interconnect hardware and software used on different physical network segments called *subnets* and forward IP packets between each of the subnets.
- To support and use DHCP service across multiple subnets, routers connecting each subnet should comply with DHCP/BOOTP relay agent capability.
- If a router cannot function as a DHCP/BOOTP relay agent, each subnet must have either its own DHCP server or another computer that can function as a relay agent on that subnet. In most cases, routers support DHCP/BOOTP relay.

IP Address assignment process

- **Transition State Diagram :**



Transition State Diagram :

- The DHCP client goes from one state to another depending on the messages it receives or sends.
- **INIT :** When the DHCP client first starts, it is in the INIT (Initialize) state. The client broadcasts a DHCPDISCOVER message, request for an IP address.
- **SELECTING :** in the reply of DHCPDISCOVER request servers offer an IP address using DHCPOFFER message. The client accepts one of them and sends a DHCPREQUEST message to the selected server. It than goes to the requesting state. If client does not get any DHCPOFFER it waits for 5 minutes and again send DHCPDISCOVER message.
- **REQUESTING :** after receiving DHCPREQUEST message server binds offered IP address with the physical address of the client and sends a DHCPACK message to the client. During this time client stays in a requesting state.

Transition State Diagram :

- **BOUND :** During this state client can use the IP address until the lease expires, when 50% of the lease expired, the client sends another DHCPREQUEST to the server for renewal of the lease period. It than goes to the renewing state. Client also can cancel the lease and go to init sate by sending DHCPRELEASE message to the server.
- **RENEWING :** Client remains in this state until it receives a DHCPACK, which renews a lease period. The client resets its timer and goes back to the bound state. If DHCPACK is not received and 87.5% of the lease period expires, the client goes to the rebinding state.
- **REBINDING :** The client remains in this state until it receives a DHCPACK, it goes to the bound state and reset the lease timer, or it receives a DHCPNACK or the lease expires it goes to the init states and tries to get another IP address.

Messages Used By DHCP

- **DHCPDISCOVER:** Client broadcast to locate available servers. It is assumed at least one of the servers will have resources to fulfill the request.
- **DHCPOFFER:** Server to client in response to DHCP Discover with offer of configuration parameters.
- **DHCPREQUEST:** Client broadcast to servers requesting offered parameters from one server and implicitly declining offers from all others.
- **DHCPRELEASE:** Client to server release network address and canceling current lease.
- **DHCPACK:** Server to client with configuration parameters, including committed network address.
- **DHCPNACK:** Server to client refusing request for configuration parameters (eg. requested network address already allocated).

Introduction to Domain Name System

- DNS stands for **Domain Name System** (or **S***ervice* or **S***erver*), an Internet service that translates domain names into IP addresses. Because domain names are alphabetic, they're easier to remember. The Internet however, is really based on IP addresses.
- Every time you use a domain name, therefore, a DNS service must translate the name into the corresponding IP address. For example, the domain name www.example.com might translate to198.105.232.4.
- The DNS system is, in fact, its own network. If one DNS server doesn't know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned

Objectives of Domain Name System

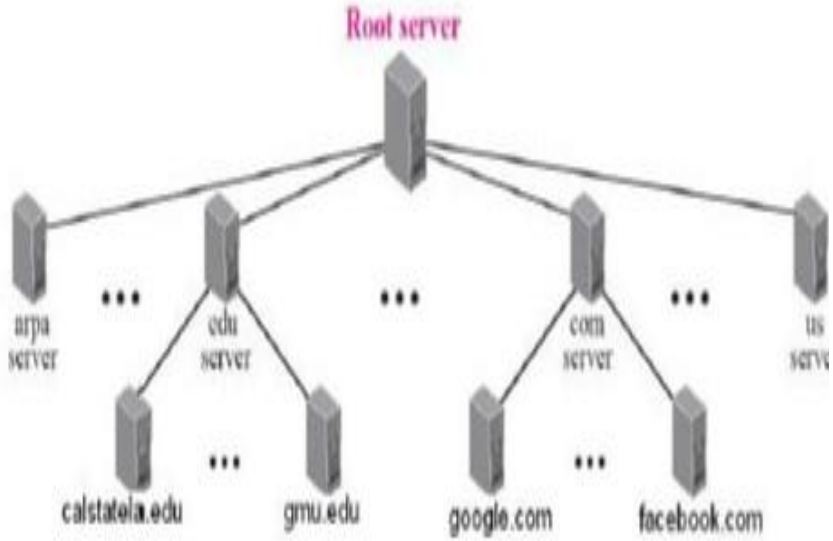
- Administrators create and assign host names to their computers without duplicating the names of other systems.
- To store the host names in a database that would be accessible by any system, anywhere on the network.
- To distribute the host name database among servers all over the network.
- To avoid creating traffic bottlenecks and a single point of failure.
- To allow name changes to be dynamically updated to all participating computers.

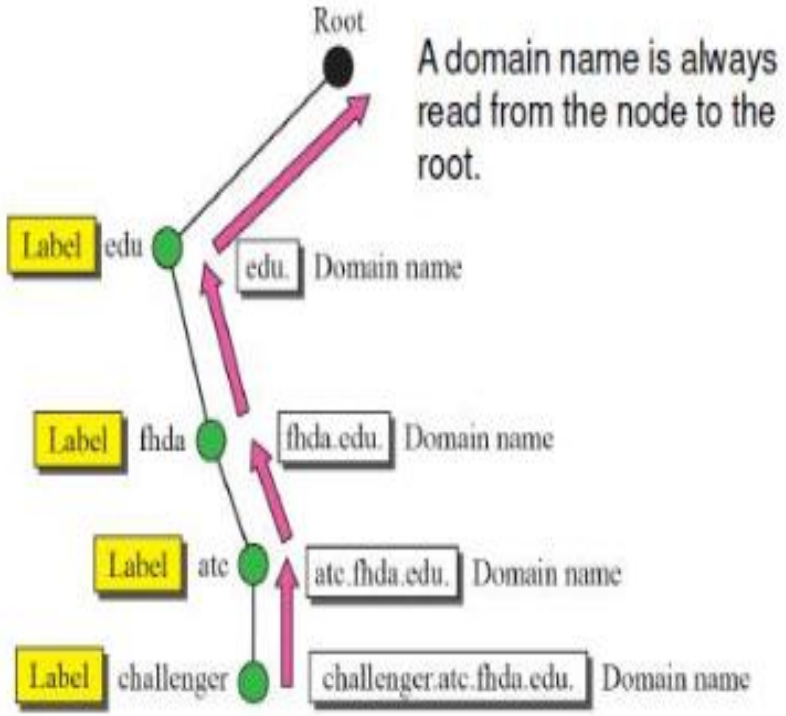
Domain Naming

- Before the DNS was developed, administrators assigned a simple host names to the computers on the networks.
- To support the network as it grew larger, developed a hierarchical name space that made it possible for individual network administrators to name their systems.
- The DNS name space is based on domains, which exist in a hierarchical structure.
- A domain is equivalent of a directory, in that it can contain either subdomains or hosts, forming a structure called DNS tree.
- every computer on the Internet is uniquely identified by a DNS Name that consists of a host name plus the names of all of its parent domains, stretching up to the root of the DNS tree, separated by periods.
- Each name can be up to 63 characters long, with a total length of 255 characters for a complete DNS name, including the host and all of its parent domains.

Domain Naming

- Thus, a DNS name is something like a postal address, in which the top–level domain is equivalent of the state, the second–level domain is the city, and the host name is the street address.





Domain Naming→ Top Level Domain

- In every DNS name, the first word on the right represents the domain at the highest level in the DNS tree, called a top–level domain.
- These top–level domains function as registrars for the domains at the second level.
- The original DNS name space called for seven top–level domains, dedicated for specific purposes.

· com – Commercial organizations
· edu – Four year, degree granting educational institutions
· gov – government institutions
· int – Organizations established by International treaty
· mil – Unites States military applications
· net – Networking organizations
· org – Noncommercial organizations.

Domain Naming→ Second Level Domain

- The registrars of the second–level domains are responsible for registering second–level domain names, in return for a subscription fee. As long as an organization continues to pay fees for its domain name, it has exclusive rights to that name.
- The domain registrar maintain records that identify the owner of each second–level domain and specify three contacts within the registrant's organization– an administrative contact, a billing contact, and a technical contact.
- In addition, the registrar must have the IP addresses of two DNS servers that function as the source for further information about the domain. This is the only information maintained by the top–level domain.
- To host a second–level domain organization must have two DNS servers.
- A DNS server is a software program that runs on a computer.
- The DNS servers identified in the top–level domains record are the authority of the second–level domain. This means that these servers are the ultimate source of information about that domain.
- When network administrators want to add a host to the network or create a new subdomain, they do so in their own DNS servers.

Domain Naming→ Subdomains

- Many of the domains on the Internet stop at two levels, meaning that the second–level domain contains only host systems.
- However, it is possible for the administrators of a second–level domain to create subdomains that form additional levels.
- Large organization use Subdomains to subdivide their networks according to geographical or organizational boundaries.
- The use of subdomain can make it easier to identify hosts on a large network, but many organizations also use them for the purpose of domain maintenance.
- The DNS servers for the top–level domain contain the addresses for each of the second–level domain's authoritative servers. In the same way, a second–level domain's servers can refer to authoritative server for third–level.

Domain Naming→ Subdomains

- To make this delegation possible, DNS servers can break up a Doman's name space into administrative units called zones.
- A domain with only two levels consists of only a single zone, which is synonymous with the domain.
- A three–level domain can be divided into multiple zones. A zone can be a continuous branch of a DNS tree, and can include domains on multiple levels.

DNS Function

- If you connect to the Internet, you use a DNS server each time you enter a server name or URL to resolve the name of the system you specified into IP address.
- There are many functions of DNS which are described below.
- **Resource Records:** DNS servers are basically database servers that store information about the hosts and subdomain for which they are responsible in resource records (RRs).
- **A (Address):** Provides a name to IP address mapping that supplies an IP address for a specific DNS name. This record type performs the primary function of the DNS, converting names to addresses.
- **MX (Mail Exchanger):** Identifies a system that will direct e–mail traffic sent to an address in the domain to the individual recipient, a mail gateway or another mail server
- **NS (Name Server) :** The name server resource record indicates the servers authoritative for the zone.

DNS Name Resolution-> Resolvers

- When you type a URL containing a DNS name (Like www.microsoft.com) into browser's address field and press the ENTER key, if you look quickly at the status bar in the lower left corner, you will see a message that says "Finding site: www.microsoft.com." In a few second you will see a message that says "Connecting to, "followed by an IP address. It is during this interval that the DNS name resolution process occurs.
- **Resolvers:** The component in the client system that generates the DNS query is called a resolver. In most cases, resolver is a simple set of library routine in the operating system that generates the queries to be sent to the DNS server, reads the response information from the server's replies, and feeds the response to the application that originally requested it. In addition, a resolver can resend a query if no reply is forthcoming after a given timeout period, and can process error messages returned by the server, such as when it fails to resolve a given name.

DNS Name Resolution-> DNS Request

- **DNS Requests:**
- A TCP/IP client usually is configured with addresses of two DNS servers which it can send queries.
- A client only need to access one DNS server, but two are usually specified to provide a backup in case one server is unavailable.
- There are two types of DNS queries
- **1. Recursive Query:** When a server receives a recursive query, it is responsible for trying to resolve the requested name and for transmitting reply back to the requester. Even if the server does not possess the required information itself, it must send its own queries to other DNS servers until it obtains the requested information or an error message stating why the information was unavailable, and must then relay the information back to the requester. The resolver in client systems nearly always sends recursive queries to DNS servers. For sending recursive query server must be configured with forwarder behavior.

DNS Name Resolution-> DNS Request

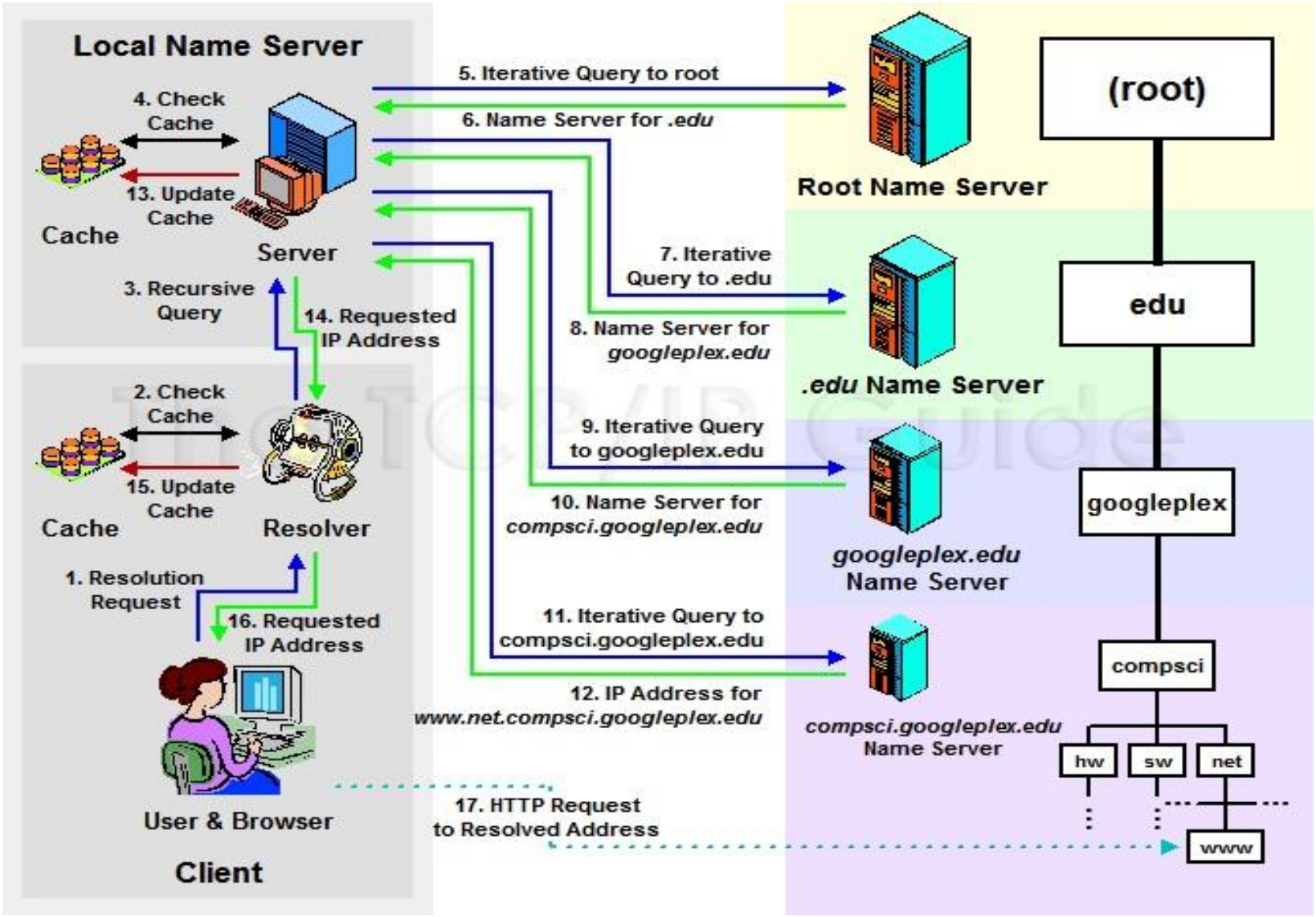
- **2. Iterative Query:** When a server receives iterative query, it can respond with information from its own database or refer the requester to another DNS server. The recipient of the query responds with the best answer it currently possesses, but is not responsible for searching for the information like recursive query does.

Root Name Server

- DNS servers that do not possess the information needed to resolve a name requested by a client send their first iterative query to one of the Internet's root name servers.
- The root name server possesses information about all of the top–level domains in the DNS name space.
- The root name servers contain the addresses of the authoritative servers for the entire top–level domain on the Internet. The root name servers are the authorities for certain top–level domains, including the country code domains, which are scattered all over the world.
- There are currently 13 root name servers, and they process millions of request each day.

Resolving the DNS Name

- The DNS name resolution process is given below.



Resolving the DNS Name

- The DNS name resolution process is given below.
- This fairly complex example illustrates a typical DNS name resolution using both iterative and recursive resolution.
- The user types in a DNS name (“www.net.compsci.googleplexplex.edu”) into a Web browser, which causes a DNS resolution request to be made from her client machine’s resolver to a local DNS name server.
- That name server agrees to resolve the name recursively on behalf of the resolver, but uses iterative requests to accomplish it.
- These requests are sent to a DNS root name server, followed in turn by the name servers for “.edu”, “googleplexplex.edu” and ‘compsci.googleplexplex.edu”.
- The IP address is then passed to the local name server and then back to the user’s resolver and finally, her Web browser software.

Resolving the DNS Name

- Now, suppose you are an employee within KDP College and one of your clients is in charge of the networking department at Googleplex. You type into your Web browser the address of this department's Web server, “www.net.compsci.googleplex.edu”. In simplified terms, the procedure would involve the following set of steps

1. Your Web browser recognizes the request for a name and invokes your local resolver, passing to it the name “www.net.compsci.googleplex.edu”.
2. The resolver checks its cache to see if it already has the address for this name. If it does, it returns it immediately to the Web browser, but in this case we are assuming that it does not. The resolver also checks to see if it has a local host table file. If so, it scans the file to see if this name has a static mapping. If so, it resolves the name using this information immediately. Again, let's assume it does not, since that would be boring.
3. The resolver generates a recursive query and sends it to “ns1.KDPCollege.com” (using that server's IP address, of course, which the resolver knows).

Resolving the DNS Name

4. The local DNS server receives the request and checks ***its*** cache. Again, let's assume it doesn't have the information needed. If it did, it would return the information, marked “non-authoritative”, to the resolver. The server also checks to see if it has in its zone resource records that can resolve “www.net.compsci.googleplex.edu”. Of course it does not, in this case, since they are in totally different domains.
5. “ns1. KDPCollege.com” generates an iterative request for the name and sends it to a root name server.
6. The root name server does not resolve the name. It returns the name and address of the name server for the “.edu” domain.
7. “ns1. KDPCollege.com” generates an iterative request and sends it to the name server for “.edu”.
8. The name server for “.edu” returns the name and address of the name server for the “googleplex.edu” domain.
9. “ns1. KDPCollege.com” generates an iterative request and sends it to the name server for “googleplex.edu”.

Resolving the DNS Name

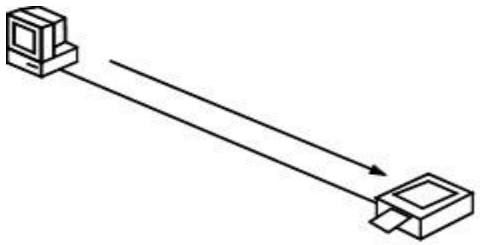
10. The name server for “googleplex.edu” consults its resource records. It sees, however, that this name is in the “compsci.googleplex.edu” subdomain, which is in a separate zone. It returns the name server for that zone.
11. “ns1. KDPCollege.com” generates an iterative request and sends it to the name server for “compsci.googleplex.edu”.
12. The name server for “compsci.googleplex.edu” is authoritative for “www.net.compsci.googleplex.edu”. It returns the IP address for that host to “ns1. KDPCollege.com”.
13. “ns1. KDPCollege.com” caches this resolution. (Note that it will probably also cache some of the other name server resolutions that it received in steps #6, #8 and #10; I have not shown these explicitly.)
14. The local name server returns the resolution to the resolver on your local machine.
15. Your local resolver also caches the information.
16. The local resolver gives the address to your browser.
17. Your browser commences an HTTP request to the Googleplex machine's IP address.

DNS Name registration

- Name resolution is the process by which IP address information for a host name is extracted from the DNS database.
- The process by which host names and their addresses are added to the database is called the Name Registration.
- Name registration refers to the process of creating new resource records on a DNS server.
- **Manual Name registration**
  - The manual name registration process, an administrator has to manually create a resource record on the server.
  - If you have a large number of hosts, manually creating resource records for all of them can be time consuming and tedious matter, even with a graphical interface.
- **Dynamic Updates**
  - As network grow larger and more complex, the biggest problem arising from manual name registration stems from the increasing use of DHCP servers to dynamically assign IP addresses to network workstation.
  - Update message is generated by domain controllers and DHCP servers and transmit to a DNS server.
  - These Update messages can modify or delete existing resource records or create new ones, based on prerequisites specified by administrator.

Locally Connected Print Devices

- A type of personal printer that is connected directly to a single desktop computer or laptop, a **local printer** typically can only be accessed by the user of the computer to which it is attached.
- The local printer is typically connected to the computer with a cable.
- It could have a USB connection (very common now) or it may connect to a parallel port (generally older printers and computers).
- It is possible, with the right software and set up, to share a local printer with other users over a network. The printer becomes a network printer but it is local to the computer to which it is directly connected.
- Printers used in this manner are generally low output machines. In general, low output capability normally means a low initial investment amount but higher supply and maintenance costs.



Setting up a local Printer

- Adding a printer to your computer involves two steps: making the connection from the computer to the printer using either a parallel or USB cable, and installing the software needed to allow your computer to communicate with the specific printer you want to use.
- Once you have physically connected the printer, you will also need to install the hardware related to that printer.
- Steps for installing local printer in windows server 2008
  1. Click the Windows 2008 "Start" button and select "Administrative Tools." Click "Print Management" to open the main configuration window.
  2. Click the "Printer Server" icon, and then click "Printers." Right– click any white space in the detail pane and select "Add Printer." The printer wizard opens.
  3. Click the "Create a port and add a printer" option to install the printer. Click "Next."
  4. Select the printer type and model in the next window. Click "Next." If you want to provide a custom name for the printer, type it in the text box and click "Finish" to install the printer.

Sharing locally attached print device

- If you have a printer attached to your computer, you can share it with anyone on the same network. It doesn't matter what type of printer you have, as long it's installed on your computer and is directly attached with a USB cable or other type of printer cable. The people you choose to share the printer with will be able to use it to print, as long as they first locate your computer on the network.
- If you want to share a printer connected directly to a Windows Server
- 2008 server, this is easy to do. First, open the server's Printers folder (from the Start menu, choose Control Panel, and then choose Printers), which lists all the installed printers. Right click the one you want to share and choose Sharing from the pop–up menu.
- The Properties dialog box for the printer will appear, with the Sharing tab activated, as shown in figure.
- On the Sharing tab, click the Share This Printer checkbox, and then assign the printer a share name, by which the client computers will recognize the printer.

Assignment

1. Explain Address Resolution Protocol.
2. Short note on RARP.
3. Explain BOOTP.
4. Write the advantages and disadvantages of DHCP.
5. Explain various IP address schemes in DHCP.
6. Explain DHCP architecture.
7. Explain DHCP Packet Format.
8. Explain DNS registration process.

Assignment

09. Discuss different messages used by DHCP.
10. Explain function of DNS server.
11. List and explain TOP level domain
12. Explain name resolver.
13. What do you mean by resolver? Explain Domain Name Resolving in detail.
14. What is network printer? Write the important of them.
15. Write a step to install network printer