Outline

- Domain Name System (DNS)
- Electronic Mail
- File Transfer Protocol (FTP)
- Hyper Text Transfer Protocol (HTTP)

Domain Name System

- The client/server programs can be divided into two categories:
  - those that can be directly used by the user (email), and
  - those that support other application programs.
- The Domain Name System (DNS) is a supporting program that is used by other programs such as e-mail.

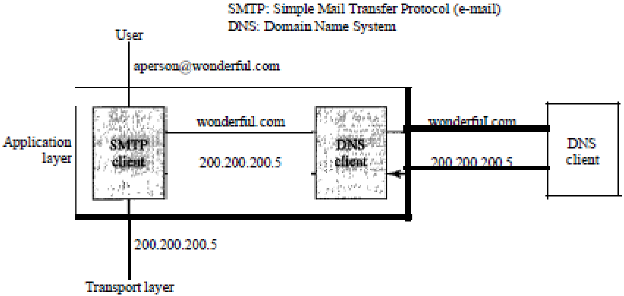

Figure: Example of using the DNS service

Domain Name System

- To identify an entity, TCPI/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet.
- However, people prefer to use names instead of numeric addresses.
- Therefore, we need a system that can map a name to an address or an address to a name.
- When the Internet was small, mapping was done by using a host file.
- The host file had only two columns: name and address.
- Every host could store the host file on its disk and update it periodically from a master host file.
- When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.

Domain Name System

- To identify an entity, TCPI/IP protocols use the IP address, which uniquely identifies the connection of a host to the Internet.
- However, people prefer to use names instead of numeric addresses.
- Therefore, we need a system that can map a name to an address or an address to a name.
- When the Internet was small, mapping was done by using a host file.
- The host file had only two columns: name and address.
- Every host could store the host file on its disk and update it periodically from a master host file.
- When a program or a user wanted to map a name to an address, the host consulted the host file and found the mapping.
- Current Scenario: it is impossible to have one single host file to relate every address with a name and vice versa. The host file would be too large to store in every host. In addition, it would be impossible to update all the host files every time there was a change.

Domain Name System

- Soluton 1:
  - Store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping.
  - Limitation: this would create a huge amount of traffic on the Internet.

Domain Name System

- Soluton 1:
  - Store the entire host file in a single computer and allow access to this centralized information to every computer that needs mapping.
  - Limitation: this would create a huge amount of traffic on the Internet.
- Solution 2:
  - Divide this huge amount of information into smaller parts and store each part on a different computer.
  - In this method, the host that needs mapping can contact the closest computer holding the needed information.
  - This method is used by the Domain Name System (DNS).

Name Space

- The names must be unique because the addresses are unique.
- A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.

Name Space

- The names must be unique because the addresses are unique.
- A name space that maps each address to a unique name can be organized in two ways: flat or hierarchical.
- Flat Name Space
  - In a flat name space, a name is assigned to an address.
  - A name in this space is a sequence of characters without structure.
  - Disadvantage: it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

Name Space

- Hierarchical Name Space
  - In a hierarchical name space, each name is made of several parts.

Name Space

- Hierarchical Name Space
  - In a hierarchical name space, each name is made of several parts.
  - The first part can define the nature of the organization.

Name Space

- Hierarchical Name Space
  - In a hierarchical name space, each name is made of several parts.
  - The first part can define the nature of the organization.
  - The second part can define the name of an organization.

Name Space

- Hierarchical Name Space
  - In a hierarchical name space, each name is made of several parts.
  - The first part can define the nature of the organization.
  - The second part can define the name of an organization.
  - The third part can define departments in the organization, and so on.
  - In this case, the authority to assign and control the name spaces can be decentralized.
  - A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.
  - The responsibility of the rest of the name can be given to the organization itself.
  - The organization can add suffixes (or prefixes) to the name to define its host or resources.

Name Space

- Hierarchical Name Space
  - In a hierarchical name space, each name is made of several parts.
  - The first part can define the nature of the organization.
  - The second part can define the name of an organization.
  - The third part can define departments in the organization, and so on.
  - In this case, the authority to assign and control the name spaces can be decentralized.
  - A central authority can assign the part of the name that defines the nature of the organization and the name of the organization.
  - The responsibility of the rest of the name can be given to the organization itself.
  - The organization can add suffixes (or prefixes) to the name to define its host or resources.
  - Example: assume two colleges and a company call one of their computers *challenger*. The first college is given a name by the central authority such as *jhda.edu*, the second college is given the name *berkeley.edu*, and the company is given the name *smart.com*. When these organizations add the name challenger to the name they have already been given, the end result is three distinguishable names: *challenger.jhda.edu*, *challenger.berkeley.edu*, and *challenger.smart.com*.

Domain Name Space

- To have a hierarchical name space, a domain name space was designed.
- In this design, the names are defined in an inverted-tree structure with the root at the top.
- The tree can have only 128 levels: level 0 (root) to level 127.

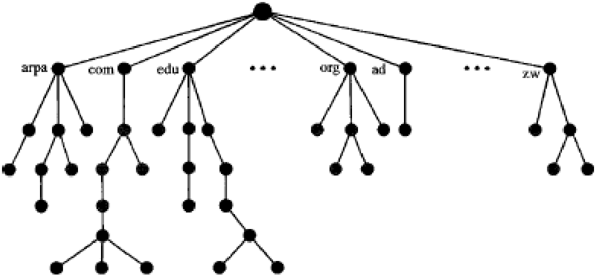

Figure: Domain name space

Domain Name Space

- Label
  - Each node in the tree has a label, which is a string with a maximum of 63 characters.
  - The root label is a null string (empty string).
  - DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.

Domain Name Space

- Label
  - Each node in the tree has a label, which is a string with a maximum of 63 characters.
  - The root label is a null string (empty string).
  - DNS requires that children of a node (nodes that branch from the same node) have different labels, which guarantees the uniqueness of the domain names.
- Domain Name
  - Each node in the tree has a domain name.
  - A full domain name is a sequence of labels separated by dots (.).
  - The domain names are always read from the node up to the root.
  - The last label is the label of the root (null).
  - This means that a full domain name always ends in a null label, which means the last character is a dot because the null string is nothing.

Domain Name Space

- Domain Name: categorized as fully qualified and partially qualified.

Domain Name Space

- Domain Name: categorized as fully qualified and partially qualified.
  - Fully Qualified Domain Name:
    - If a label is terminated by a null string, it is called a fully qualified domain name (FQDN).
    - An FQDN is a domain name that contains the full name of a host.
    - It contains all labels, from the most specific to the most general, that uniquely define the name of the host.
    - Example: the domain name *challenger.ate.tbda.edu* is the FQDN of a computer named *challenger* installed at the Advanced Technology Center (ATC) at De Anza College.
    - A DNS server can only match an FQDN to an address.
    - Note: the name must end with a null label, but because null means nothing, the label ends with a dot (.).

Domain Name Space

- Partially Qualified Domain Name:
  - If a label is not terminated by a null string, it is called a partially qualified domain name (PQDN).
  - A PQDN starts from a node, but it does not reach the root.
  - It is used when the name to be resolved belongs to the same site as the client.
  - The resolver can supply the missing part, called the *suffix*, to create an FQDN.
  - Example: if a user at the *jhda.edu* site wants to get the IP address of the challenger computer, he or she can define the partial name *challenger*.
  - The DNS client adds the suffix *atc.jhda.edu* before passing the address to the DNS server.
  - The DNS client normally holds a list of suffixes.
  - The following can be the list of suffixes at De Anza College: *atc.fhda.edu*, *fhda.edu*, and *null*.
  - The null suffix defines nothing.
  - This suffix is added when the user defines an FQDN.

Domain Name Space

- Domain
  - A domain is a subtree of the domain name space.
  - The name of the domain is the domain name of the node at the top of the subtree.
  - Note: a domain may itself be divided into domains (or subdomains).

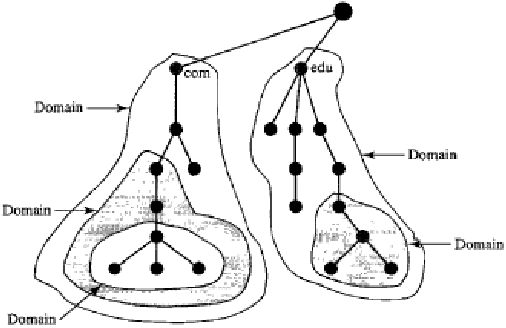

Figure: Domains

Distribution of Name Space

- The information contained in the domain name space must be stored.
- It is very inefficient and also unreliable to have just one computer store such a huge amount of information.
- It is inefficient because responding to requests from all over the world places a heavy load on the system.
- It is not reliable because any failure makes the data inaccessible.

Distribution of Name Space

- The information contained in the domain name space must be stored.
- It is very inefficient and also unreliable to have just one computer store such a huge amount of information.
- It is inefficient because responding to requests from all over the world places a heavy load on the system.
- It is not reliable because any failure makes the data inaccessible.
- Hierarchy of Name Servers
  - The solution to above problems is to distribute the information among many computers called DNS servers.
  - One way to do this is to divide the whole space into many domains based on the first level.
  - In other words, we let the root stand alone and create as many domains (subtrees) as there are first-level nodes.
  - Because a domain created in this way could be very large, DNS allows domains to be divided further into smaller domains (subdomains).
  - Each server can be responsible (authoritative) for either a large or a small domain.
  - In other words, we have a hierarchy of servers in the same way that we have a hierarchy of names.

Distribution of Name Space

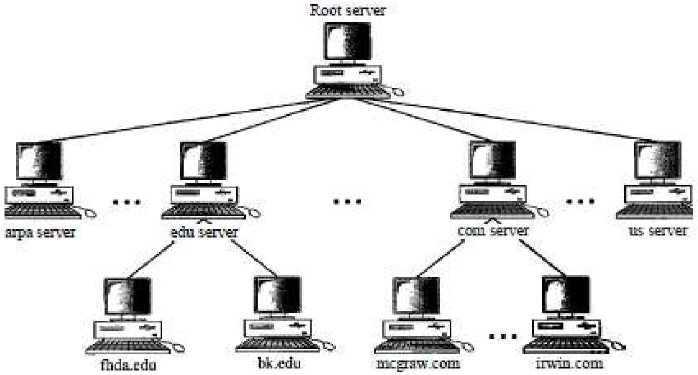

Figure: Hierarchy of name servers

Distribution of Name Space

- Zone
  - The complete domain name hierarchy cannot be stored on a single server, it is divided among many servers.
  - Zone: a contiguous part of the entire tree.

Distribution of Name Space

- Zone
  - The complete domain name hierarchy cannot be stored on a single server, it is divided among many servers.
  - Zone: a contiguous part of the entire tree.
  - If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the domain and the zone refer to the same thing.
  - The server makes a database called a zone file and keeps all the information for every node under that domain.

Distribution of Name Space

- Zone
  - The complete domain name hierarchy cannot be stored on a single server, it is divided among many servers.
  - Zone: a contiguous part of the entire tree.
  - If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the domain and the zone refer to the same thing.
  - The server makes a database called a zone file and keeps all the information for every node under that domain.
  - If a server divides its domain into subdomains and delegates part of its authority to other servers, domain and zone refer to different things.
  - The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers.

Distribution of Name Space

- Zone
  - The complete domain name hierarchy cannot be stored on a single server, it is divided among many servers.
  - Zone: a contiguous part of the entire tree.
  - If a server accepts responsibility for a domain and does not divide the domain into smaller domains, the domain and the zone refer to the same thing.
  - The server makes a database called a zone file and keeps all the information for every node under that domain.
  - If a server divides its domain into subdomains and delegates part of its authority to other servers, domain and zone refer to different things.
  - The information about the nodes in the subdomains is stored in the servers at the lower levels, with the original server keeping some sort of reference to these lower-level servers.
  - A server can also divide part of its domain and delegate responsibility but still keep part of the domain for itself.
  - In this case, its zone is made of detailed information for the part of the domain that is not delegated and references to those parts that are delegated.

Distribution of Name Space

- Zone

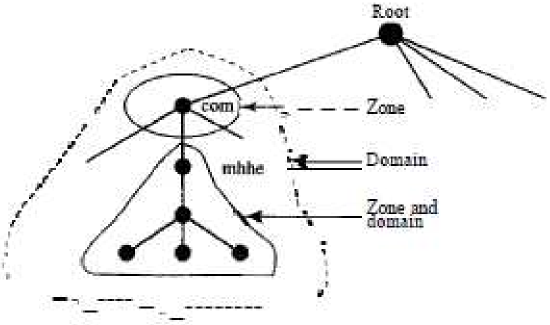

Figure: Zones and domains

Distribution of Name Space

- Root Server
  - A root server is a server whose zone consists of the whole tree.
  - A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
  - There are several root servers, each covering the whole domain name space.
  - The servers are distributed all around the world.

Distribution of Name Space

- Root Server
  - A root server is a server whose zone consists of the whole tree.
  - A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
  - There are several root servers, each covering the whole domain name space.
  - The servers are distributed all around the world.
  - DNS defines two types of servers: primary and secondary.

Distribution of Name Space

- Root Server
  - A root server is a server whose zone consists of the whole tree.
  - A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
  - There are several root servers, each covering the whole domain name space.
  - The servers are distributed all around the world.
  - DNS defines two types of servers: primary and secondary.
  - A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.

Distribution of Name Space

- Root Server
  - A root server is a server whose zone consists of the whole tree.
  - A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
  - There are several root servers, each covering the whole domain name space.
  - The servers are distributed all around the world.
  - DNS defines two types of servers: primary and secondary.
  - A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.
  - A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.

Distribution of Name Space

- Root Server
  - A root server is a server whose zone consists of the whole tree.
  - A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
  - There are several root servers, each covering the whole domain name space.
  - The servers are distributed all around the world.
  - DNS defines two types of servers: primary and secondary.
  - A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.
  - A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.
  - Idea: put the secondary server at a lower level of authority but to create redundancy for the data (if one server fails, the other can continue serving clients).

Distribution of Name Space

- Root Server
  - A root server is a server whose zone consists of the whole tree.
  - A root server usually does not store any information about domains but delegates its authority to other servers, keeping references to those servers.
  - There are several root servers, each covering the whole domain name space.
  - The servers are distributed all around the world.
  - DNS defines two types of servers: primary and secondary.
  - A primary server is a server that stores a file about the zone for which it is an authority. It is responsible for creating, maintaining, and updating the zone file. It stores the zone file on a local disk.
  - A secondary server is a server that transfers the complete information about a zone from another server (primary or secondary) and stores the file on its local disk. The secondary server neither creates nor updates the zone files. If updating is required, it must be done by the primary server, which sends the updated version to the secondary.
  - Idea: put the secondary server at a lower level of authority but to create redundancy for the data (if one server fails, the other can continue serving clients).
  - Note: a server can be a primary server for a specific zone and a secondary server for another zone.

DNS in the Internet

- In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and inverse domain.

DNS in the Internet

- In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and inverse domain.
- Generic Domains: The generic domains define registered hosts according to their generic behavior. Each node in the tree defines a domain, which is an index to the domain name space database.

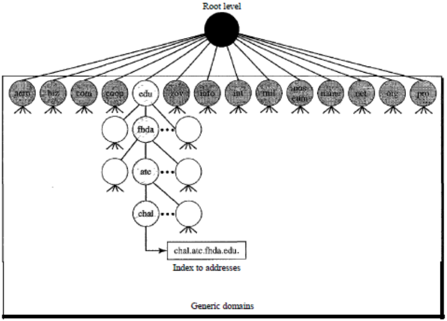

Figure: Generic domains

DNS in the Internet

- Generic Domains

Label	Description
aero	Airlines and aerospace companies
biz	Businesses or firms
com	Commercial organizations
coop	Cooperative business organizations
edu	Educational institutions
gov	Governmental institutions
info	Information service providers
int	International organizations
mil	Military groups
museum	Museums and other nonprofit organizations
name	Personal names (individuals)
net	Network support centers
org	Nonprofit organizations
pro	Professional individual organizations

DNS in the Internet

- Country Domains
  - The country domains section uses two-character country abbreviations (e.g., us for United States).
  - Second labels can be organizational, or they can be more specific, national designations.

DNS in the Internet

- Country Domains
  - The country domains section uses two-character country abbreviations (e.g., us for United States).
  - Second labels can be organizational, or they can be more specific, national designations.
- Inverse Domains
  - The inverse domain is used to map an address to a name.
  - This may happen when a server has received a request from a client to do a task.
  - Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed.
  - The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list.
  - This type of query is called an inverse or pointer (PTR) query.

DNS in the Internet

- Country Domains
  - The country domains section uses two-character country abbreviations (e.g., us for United States).
  - Second labels can be organizational, or they can be more specific, national designations.
- Inverse Domains
  - The inverse domain is used to map an address to a name.
  - This may happen when a server has received a request from a client to do a task.
  - Although the server has a file that contains a list of authorized clients, only the IP address of the client (extracted from the received IP packet) is listed.
  - The server asks its resolver to send a query to the DNS server to map an address to a name to determine if the client is on the authorized list.
  - This type of query is called an inverse or pointer (PTR) query.
  - To handle a pointer query, the inverse domain is added to the domain name space with the first-level node called arpa.
  - The second level is also one single node named in-addr (for inverse address).
  - The rest of the domain defines IP addresses.

DNS in the Internet

- Inverse Domains
  - The servers that handle the inverse domain are also hierarchical.
  - This means the netid part of the address should be at a higher level than the subnetid part, and the subnetid part higher than the hostid part.
  - In this way, a server serving the whole site is at a higher level than the servers serving each subnet.
  - This configuration makes the domain look inverted when compared to a generic or country domain.
  - Example: an IP address such as *132.34.45.121* (a class B address with netid *132.34*) is read as *121.45.34.132.in-addr.arpa*.

Resolution

- Mapping a name to an address or an address to a name is called name-address resolution.

Resolution

- Mapping a name to an address or an address to a name is called name-address resolution.
- Resolver
  - DNS is designed as a client/server application.
  - A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver.
  - The resolver accesses the closest DNS server with a mapping request.
  - If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.
  - After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.

Resolution

- Mapping a name to an address or an address to a name is called name-address resolution.
- Resolver
  - DNS is designed as a client/server application.
  - A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver.
  - The resolver accesses the closest DNS server with a mapping request.
  - If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.
  - After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.
  - Mapping names to addresses: the resolver gives a domain name to the server and asks for the corresponding address, in which case, the server checks the generic domains or the country domains to find the mapping.

Resolution

- Mapping a name to an address or an address to a name is called name-address resolution.
- Resolver
  - DNS is designed as a client/server application.
  - A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver.
  - The resolver accesses the closest DNS server with a mapping request.
  - If the server has the information, it satisfies the resolver; otherwise, it either refers the resolver to other servers or asks other servers to provide the information.
  - After the resolver receives the mapping, it interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it.
  - Mapping names to addresses: the resolver gives a domain name to the server and asks for the corresponding address, in which case, the server checks the generic domains or the country domains to find the mapping.
  - Mapping addresses to names: A client can send an IP address to a server to be mapped to a domain name.

Resolution

- Recursive Resolution
  - The client (resolver) can ask for a recursive answer from a name server.
  - If the server is the authority for the domain name, it checks its database and responds.

Resolution

- Recursive Resolution
  - The client (resolver) can ask for a recursive answer from a name server.
  - If the server is the authority for the domain name, it checks its database and responds.
  - If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response.

Resolution

- Recursive Resolution
  - The client (resolver) can ask for a recursive answer from a name server.
  - If the server is the authority for the domain name, it checks its database and responds.
  - If the server is not the authority, it sends the request to another server (the parent usually) and waits for the response.
  - If the parent is the authority, it responds; otherwise, it sends the query to yet another server.
  - When the query is finally resolved, the response travels back until it finally reaches the requesting client.

Resolution

- Recursive Resolution

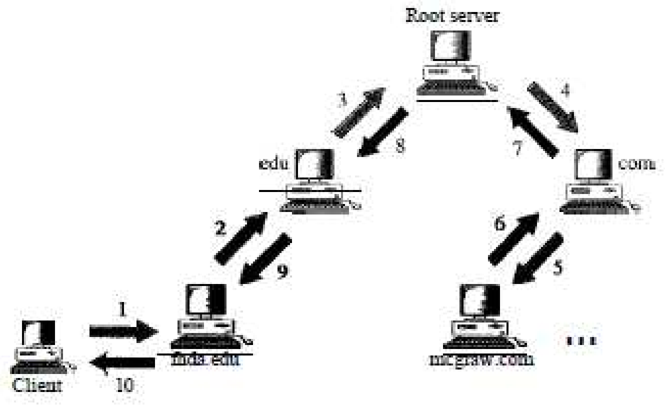

Figure: Recursive Resolution

Resolution

- Iterative Resolution
  - If the client does not ask for a recursive answer, the mapping can be done iteratively.

Resolution

- Iterative Resolution
  - If the client does not ask for a recursive answer, the mapping can be done iteratively.
  - If the server is an authority for the name, it sends the answer.

Resolution

- Iterative Resolution
  - If the client does not ask for a recursive answer, the mapping can be done iteratively.
  - If the server is an authority for the name, it sends the answer.
  - If it is not, it returns (to the client) the IP address of the server that it thinks can resolve the query.
  - The client is responsible for repeating the query to this second server.

Resolution

- Iterative Resolution
  - If the client does not ask for a recursive answer, the mapping can be done iteratively.
  - If the server is an authority for the name, it sends the answer.
  - If it is not, it returns (to the client) the IP address of the server that it thinks can resolve the query.
  - The client is responsible for repeating the query to this second server.
  - If the newly addressed server can resolve the problem, it answers the query with the IP address; otherwise, it returns the IP address of a new server to the client.
  - Now the client must repeat the query to the third server.

Resolution

- Iterative Resolution

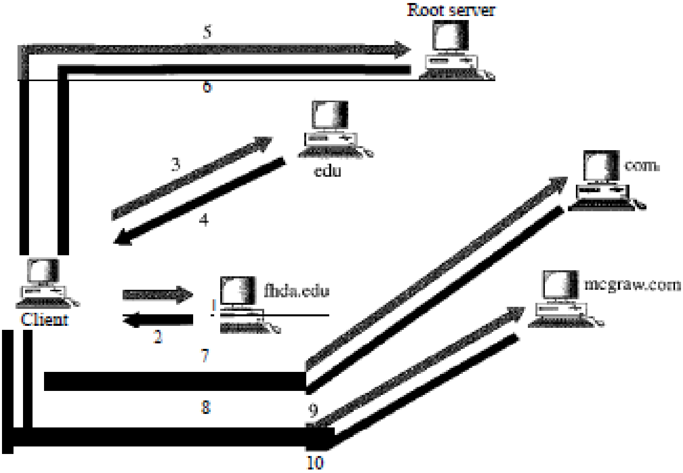

Figure: Iterative Resolution

Resolution

- Caching
  - Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
  - Reduction of this search time would increase efficiency (DNS uses caching).

Resolution

- Caching
  - Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
  - Reduction of this search time would increase efficiency (DNS uses caching).
  - When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.

Resolution

- Caching
  - Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
  - Reduction of this search time would increase efficiency (DNS uses caching).
  - When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
  - If the same or another client asks for the same mapping, it can check its cache memory and solve the problem.

Resolution

- Caching
  - Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
  - Reduction of this search time would increase efficiency (DNS uses caching).
  - When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
  - If the same or another client asks for the same mapping, it can check its cache memory and solve the problem.
  - To inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as unauthoritative.

Resolution

- Caching
  - Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
  - Reduction of this search time would increase efficiency (DNS uses caching).
  - When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
  - If the same or another client asks for the same mapping, it can check its cache memory and solve the problem.
  - To inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as unauthoritative.
  - Problem: If a server caches a mapping for a long time, it may send an outdated mapping to the client.

Resolution

- Caching
  - Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
  - Reduction of this search time would increase efficiency (DNS uses caching).
  - When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
  - If the same or another client asks for the same mapping, it can check its cache memory and solve the problem.
  - To inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as unauthoritative.
  - Problem: If a server caches a mapping for a long time, it may send an outdated mapping to the client.
  - Solution 1: the authoritative server always adds information to the mapping called time-to-live (TTL). It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server.

Resolution

- Caching
  - Each time a server receives a query for a name that is not in its domain, it needs to search its database for a server IP address.
  - Reduction of this search time would increase efficiency (DNS uses caching).
  - When a server asks for a mapping from another server and receives the response, it stores this information in its cache memory before sending it to the client.
  - If the same or another client asks for the same mapping, it can check its cache memory and solve the problem.
  - To inform the client that the response is coming from the cache memory and not from an authoritative source, the server marks the response as unauthoritative.
  - Problem: If a server caches a mapping for a long time, it may send an outdated mapping to the client.
  - Solution 1: the authoritative server always adds information to the mapping called time-to-live (TTL). It defines the time in seconds that the receiving server can cache the information. After that time, the mapping is invalid and any query must be sent again to the authoritative server.
  - Solution 2: DNS requires that each server keep a TTL counter for each mapping it caches. The cache memory must be searched periodically, and those mappings with an expired TTL must be purged.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.
- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.
- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating.
- The size of today's Internet does not allow for this kind of manual operation.
- The DNS master file must be updated dynamically.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.
- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating.
- The size of today's Internet does not allow for this kind of manual operation.
- The DNS master file must be updated dynamically.
- The Dynamic Domain Name System (DDNS) was devised to respond to this need.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.
- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating.
- The size of today's Internet does not allow for this kind of manual operation.
- The DNS master file must be updated dynamically.
- The Dynamic Domain Name System (DDNS) was devised to respond to this need.
- In DDNS, when a binding between a name and an address is determined, the information is sent, usually by DHCP to a primary DNS server.
- The primary server updates the zone.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.
- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating.
- The size of today's Internet does not allow for this kind of manual operation.
- The DNS master file must be updated dynamically.
- The Dynamic Domain Name System (DDNS) was devised to respond to this need.
- In DDNS, when a binding between a name and an address is determined, the information is sent, usually by DHCP to a primary DNS server.
- The primary server updates the zone.
- The secondary servers are notified either actively or passively.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.
- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating.
- The size of today's Internet does not allow for this kind of manual operation.
- The DNS master file must be updated dynamically.
- The Dynamic Domain Name System (DDNS) was devised to respond to this need.
- In DDNS, when a binding between a name and an address is determined, the information is sent, usually by DHCP to a primary DNS server.
- The primary server updates the zone.
- The secondary servers are notified either actively or passively.
- In active notification, the primary server sends a message to the secondary servers about the change in the zone.

Dynamic DNS

- When the DNS was designed, no one predicted that there would be so many address changes.
- In DNS, when there is a change, such as adding a new host, removing a host, or changing an IP address, the change must be made to the DNS master file.
- These types of changes involve a lot of manual updating.
- The size of today's Internet does not allow for this kind of manual operation.
- The DNS master file must be updated dynamically.
- The Dynamic Domain Name System (DDNS) was devised to respond to this need.
- In DDNS, when a binding between a name and an address is determined, the information is sent, usually by DHCP to a primary DNS server.
- The primary server updates the zone.
- The secondary servers are notified either actively or passively.
- In active notification, the primary server sends a message to the secondary servers about the change in the zone.
- In passive notification, the secondary servers periodically check for any changes.

DNS

- Note: DNS uses the services of UDP or TCP using the well-known port 53.