# Protocols & Application

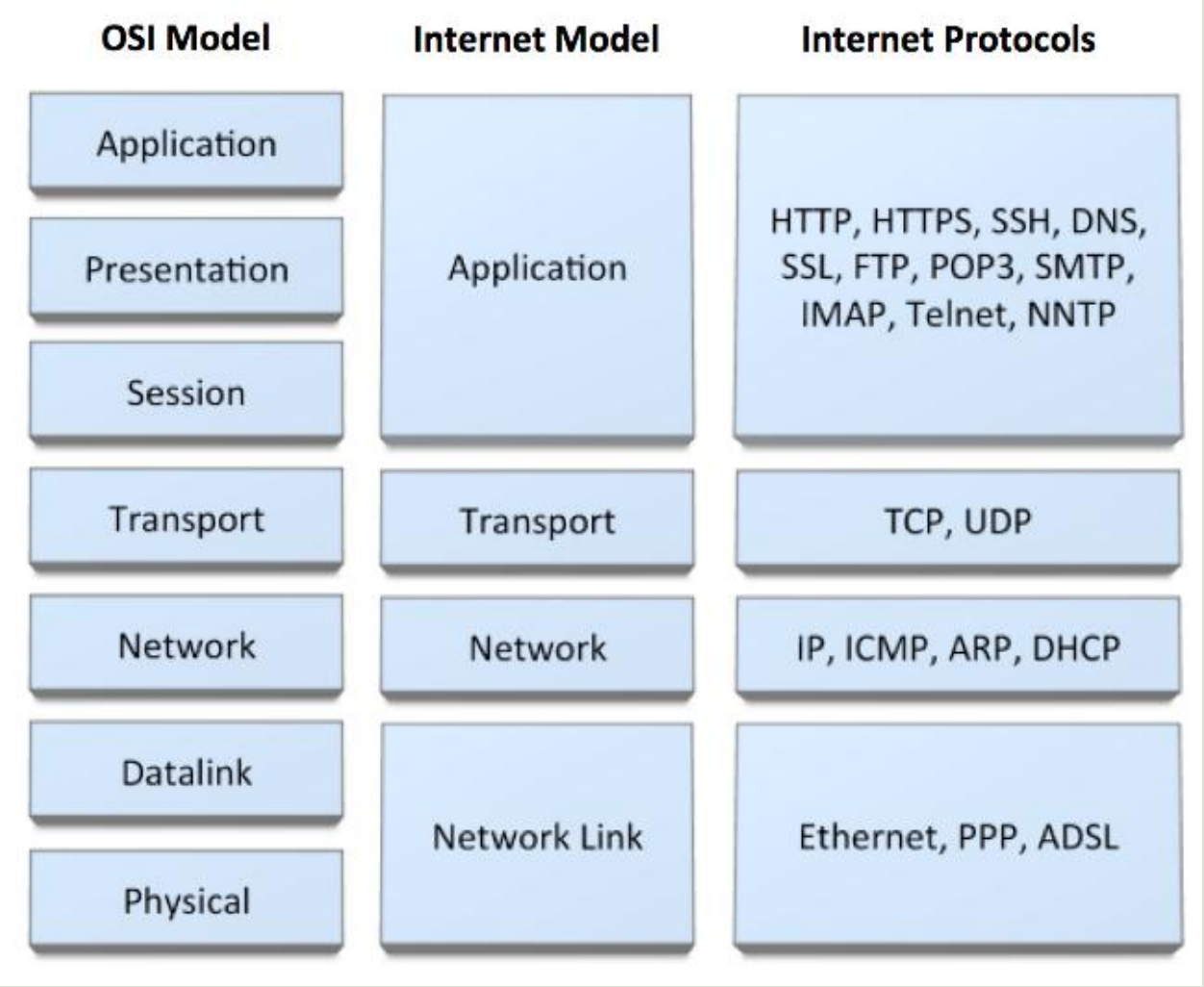
CN#3

# What is Protocol?

**Protocol**: In computer networks, communication occurs between entities in different systems. However, two entities can not simply send bit streams to each other and expect to be understood. For communication to occur, the entities must agree on a protocol. A protocol is a set of rules that govern data communications. A protocol defines what is communicated, how it is communicated, and when it is communicated.

Example: HTTP, FTP, TCP,IP etc.

# Different Layers Protocol



# Protocol Types

- **PUSH protocol:** In push protocols, the client opens a connection to the server and keeps it constantly active. The server will send (push) all new events to the client using that single always-on connection. In other words, the server PUSHes the new events to the client. Example: SMTP

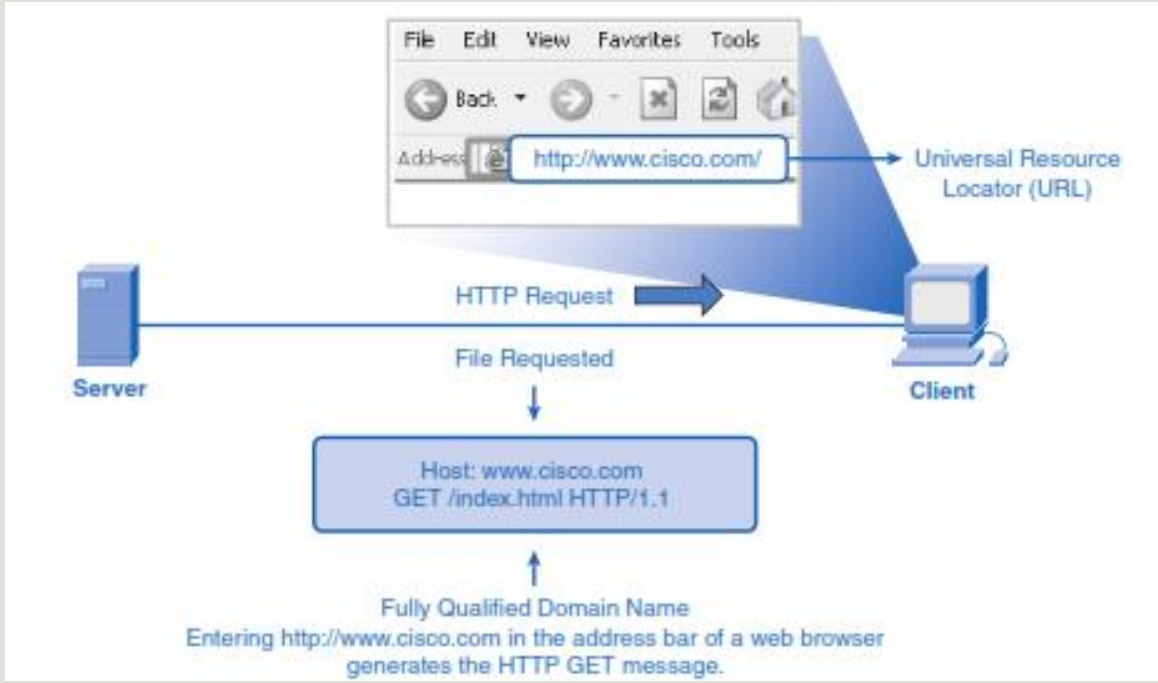
- **PULL protocol:** In pull protocols, the client periodically connects to the server, checks for and gets (pulls) recent events and then closes the connection and disconnects from the server. The client repeats this whole procedure to get updated about new events. In this mode, the clients periodically PULLs the new events from the server. Example: HTTP

# Application Layer Protocol: HTTP

- HTTP (Hypertext Transfer Protocol) is used across the world wide web (www) for data transfer and is one of the most used application protocols.
- HTTP specifies a request/response protocol. When a client, typically a web browser, sends a request message to a server, the HTTP protocol defines the message types the client uses to request the web page and the message types the server uses to respond
- The three common message types are:
  - GET
  - POST
  - PUT

# Application Layer Protocol: HTTP

GET: It is a client request for data. A web browser sends the GET message to request pages from a web server. As shown in Figure, when the server receives the GET request, it responds with a status line, such as HTTP/1.1 200 OK, and a message of its own, the body of which can be the requested file, an error message, or some other information.



# Application Layer Protocol: HTTP

- POST and PUT are used to send messages that upload data to the web server.
- For example, when the user enters data into a form embedded in a web page, POST includes the data in the message sent to the server.
- PUT uploads resources or content to the web server.

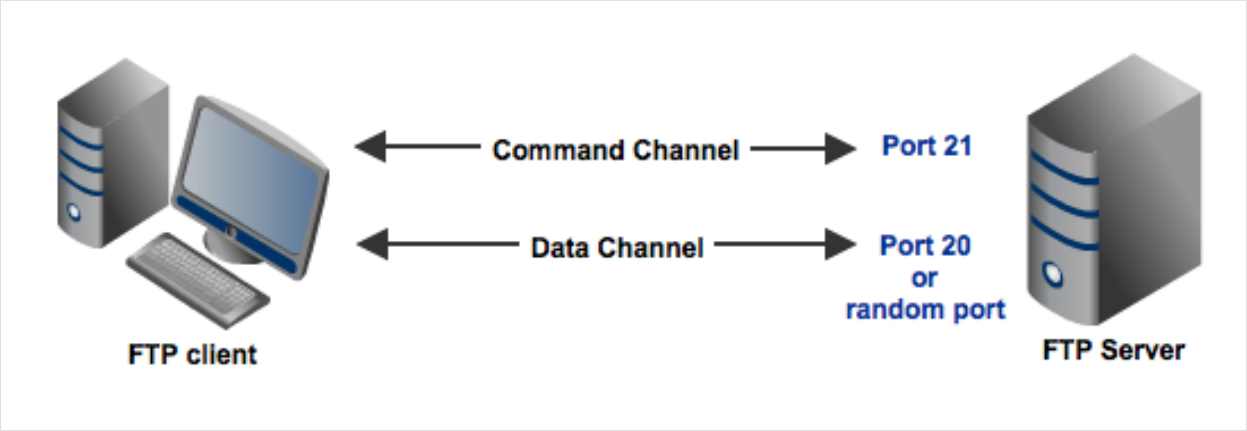
# Application Layer Protocol: HTTP runs on TCP

- The transfer protocol used throughout the World Wide Web is HTTP (Hyper Text Transfer Protocol). TCP provides a reliable data transfer service to HTTP, using TCP port 80. This implies that each HTTP request message emitted by a client process eventually arrives in tact at the server; similarly, each HTTP response message emitted by the server process eventually arrives in tact at the client. HTTP need not worry about lost data, or the details of how TCP recovers from loss or reordering of data within the network. That is the job of TCP and the protocols in the lower layers of the protocol stack

# Application Layer Protocol: FTP

- FTP is another commonly used application layer protocol. FTP was developed to allow file transfers between a client and a server. An FTP client is an application that runs on a computer that is used to push and pull files from a server running the FTP daemon (FTPd).
- To successfully transfer files, FTP requires two connections between the client and the server: one for commands and replies, and the other for the actual file transfer.
- The client establishes the first connection to the server on TCP port 21. This connection is used for control traffic, consisting of client commands and server replies.
- The client establishes the second connection to the server over TCP port 20. This connection is for the actual file transfer and is created every time a file is transferred.
- The file transfer can happen in either direction, as shown in Figure. The client can download (pull) a file from the server or upload (push) a file to the server.

# Application Layer Protocol: FTP



# Application Layer Protocol: DNS

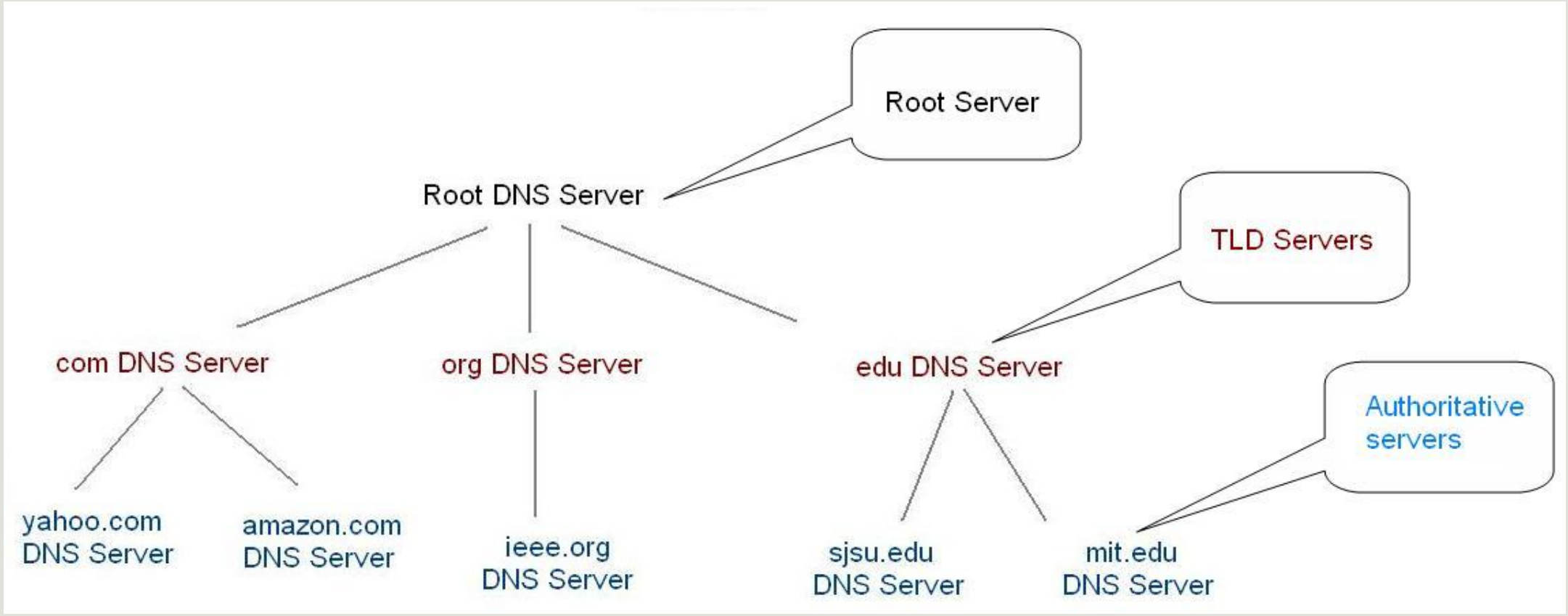
Domain Name Service (DNS):

1. DNS provides a core Internet function, translating hostnames to their underlying IP addresses, for user applications and other software in the Internet. The DNS protocol defines an automated service that matches resource names with the required numeric network address

- DNS uses a large number of name servers, organized in a hierarchical fashion and distributed around the world. No one name server has all of the mappings for all of the hosts in the internet. Instead, the mappings are distributed across the name servers. There are three types of name servers:

1. **Root name servers**
2. **Local  name server**
3. **Authoritative name servers**

# DNS Hierarchy



# DNS cont.

1. **Root name servers:** A root server contains information about the root and top-level domains, When a local name server cannot immediately satisfy a query from a host (because it does not have a record for the hostname being requested), the local name server behaves as a DNS client and queries one of the root name servers.
2. **Local name server**: Each ISP - such as a university, an academic department, an employee's company or a residential ISP - has a local name server (also called a default name server). When a host issues a DNS query message, the message is first sent to the host's local name server. The IP address of the local name server is typically configured by hand in a host.
3. **Authoritative name servers:** Every host is registered with an authoritative name server. Typically, the authoritative name server for a host is a name server in the host's local ISP. Many name servers act as both local and authoritative name servers.

# Internet Domain Names

- As an example, consider a namespace with names of the form:
  - ***local. site***

where ***site*** is the site name authorized by the central authority***, local*** is the part of a name controlled by the site, and (".") is a delimiter used to separate them.

- adding a ***group*** subdivision to names already partitioned by site produces the following name syntax:

***local. group. site***

# Internet Domain Names

Example: ***cs .purdue . edu***

contains three labels: ***cs, purdue,*** and ***edu.*** Any suffix of a label in a domain name is also called a ***domain.*** In the above example the lowest level domain is ***cs .purdue. edu,*** (the domain name for the Computer Science Department at Purdue University), the second level domain is ***purdue. edu*** (the domain name for Purdue University), and the top-level domain is ***edu*** (the domain name for educational institutions). As the example shows, domain names are written with the local label first and the top domain last.

| Domain Name | Meaning |
|---|---|
| COM | Commercial organizations |
| EDU | Educational institutions (4-year) |
| GOV | Government institutions |
| MIL | Military groups |
| NET | Major network support centers |
| ORG | Organizations other than those above |
| ARPA | Temporary ARPANET domain (obsolete) |
| INT | International organizations |
| *country code* | Each country (geographic scheme) |

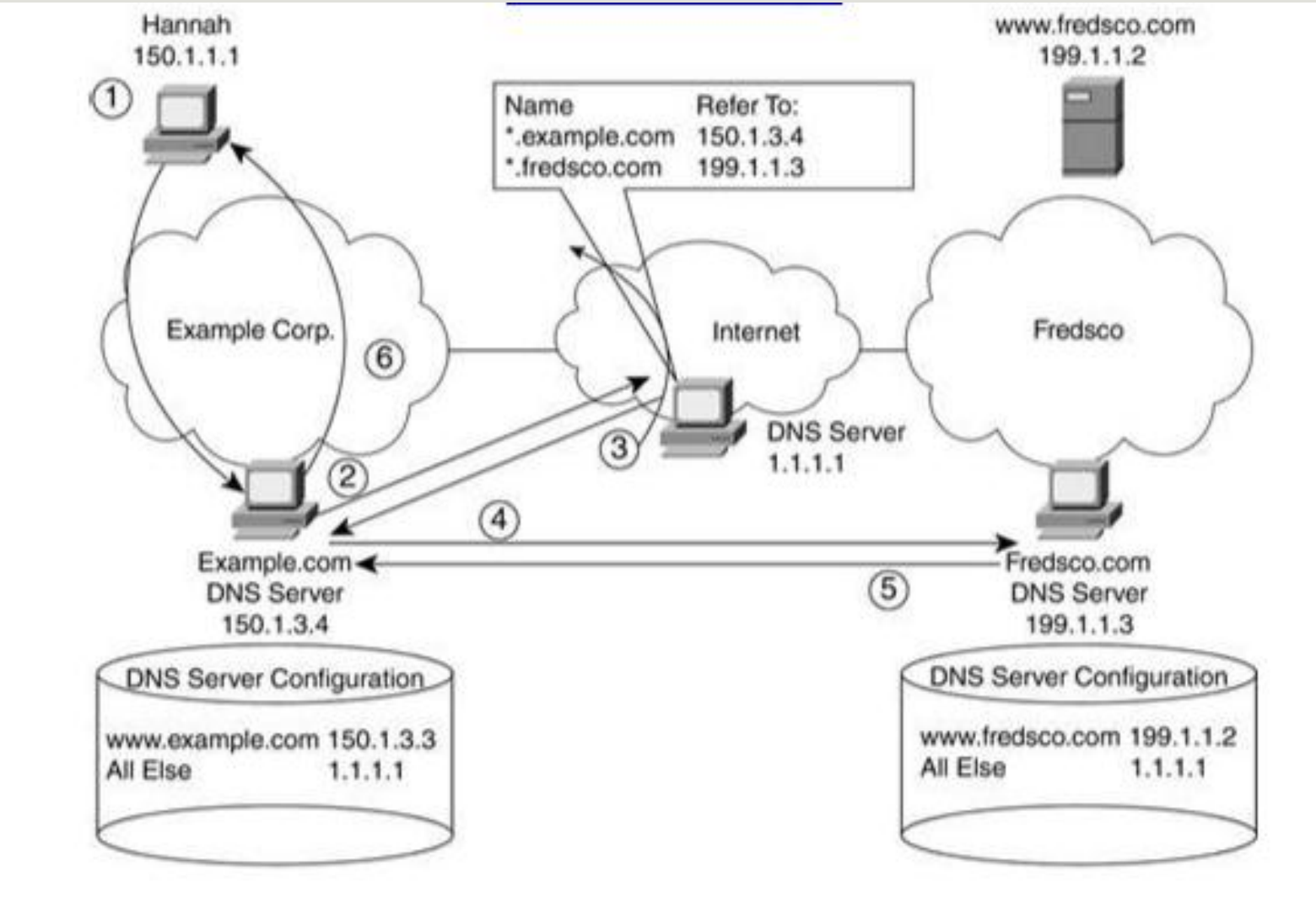
# DNS: How it works?
# Example: Hannah want to connect at www.fredsco.com

1. Hannah opens a browser, types in www.fredsco.com, and sends a DNS resolution request to her DNS (example.com).
2. example.com's DNS doesn't know the name http://www.fredsco.com. However, the DNS now has some configuration that tells it that if it doesn't know the name, it should ask the DNS server at IP address 1.1.1.1 (Root server).
3. The DNS server at 1.1.1.1 has a table that lists the IP addresses of a bunch of name servers. DNS server 1.1.1.1 knows that for all names that end in “example.com,“ DNS 150.1.3.4 can resolve the names. It also knows that for all names that end in "fredsco.com," DNS 199.1.1.3 can resolve the names. Finally, DNS server 1.1.1.1 can direct each request to the right name server. This DNS sends a message back to the example.com DNS, referring it to the DNS at 199.1.1.3.

# DNS: How it works?

4. The example.com DNS repeats the name resolution request, now sending the request to the DNS server at Fredsco (199.1.1.3).
5. The Fredsco DNS gets the request, and it knows the name and IP address. It sends a reply to the requesting host, namely, example.com's DNS server.
6. Finally, the example.com DNS server replies to Hannah, telling her that www.fredsco.com resolves to IP address 199.1.1.2.

# DNS: How it works?



# Different types of DNS queries

DNS queries can be classified according the manner in which a complete request is processed. Generally queries can be classified as follows.

1. **Recursive query:** A recursive query is a kind of query, in which the DNS server, who received the sender’s query will do all the job of fetching the answer, and giving it back to the sender. During this process, the DNS server might also query other DNS server's in the internet of the sender’s behalf, for the answer.
2. **Iterative query OR Nonrecursive query:** In an iterative query, the name server, will not go and fetch the complete answer for the sender’s query, but will give back a referral to other DNS server's, which might have the answer.

# Recursive query/ Iterative query

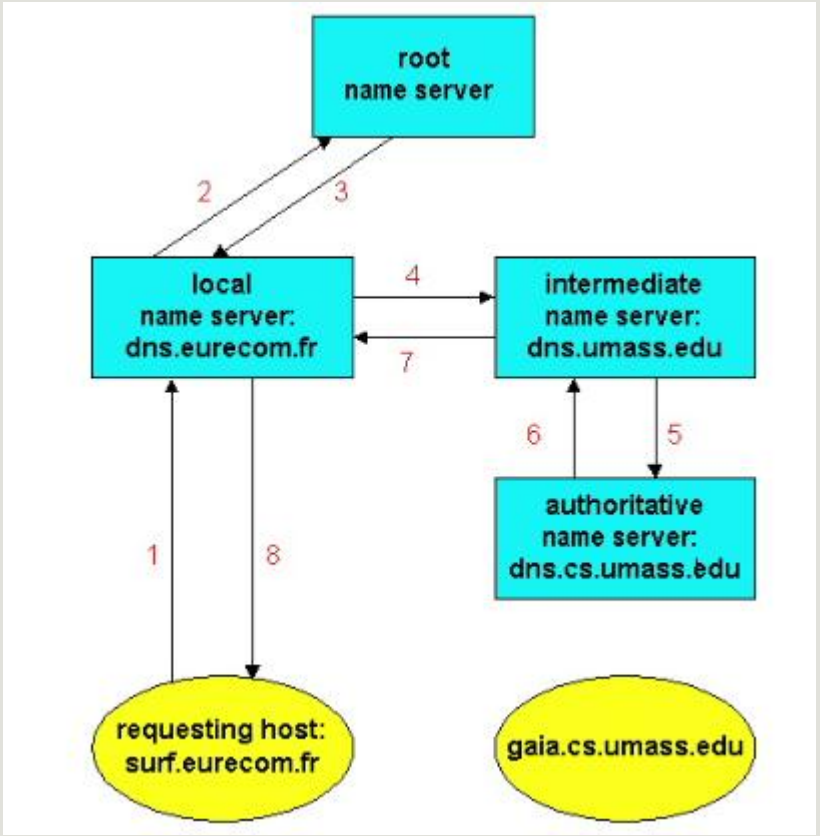



**Fig: A query chain with iterative queries.**

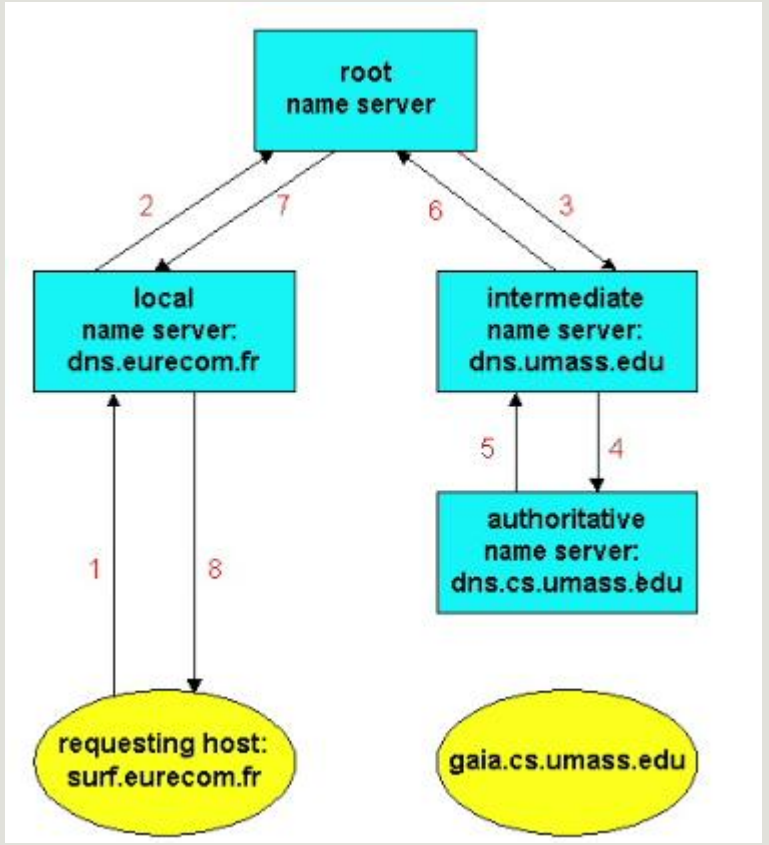



**Fig: A query chain with recursive queries.**