**UNIT 5**

**APPLICATION LAYER**

Domain Name System - File Transfer – Web Services and SNMP - HTTP - Electronic Mail (SMTP, POP3, IMAP, MIME).

|  |  |
| --- | --- |
| **Services** | **Protocols** |
| * Domain Name System * File Transfer * Web Services | * SNMP * HTTP * Electronic Mail   (SMTP, POP3, IMAP, MIME) |

**Introduction:**

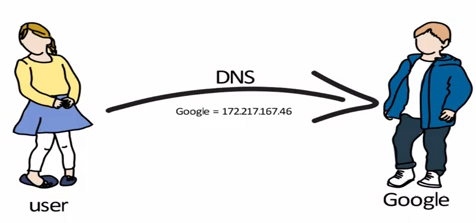
* The application layer **is responsible for providing services to the user.**
* It **provides user interfaces and support for services** such as
  + Electronic mail
  + File access and transfer
  + Access to system resources
  + Surfing the World Wide Web
  + Network management.
* There are several applications in the application layer of Internet model **that follow client/server paradigm.**
* The client/server programs can be divided into two categories:
  + **Directly used by the user,** such as **E-mail**
  + **Support other application programs** such as **DNS**
  1. **Domain Name System** **(Binding between the names and IP addresses)**

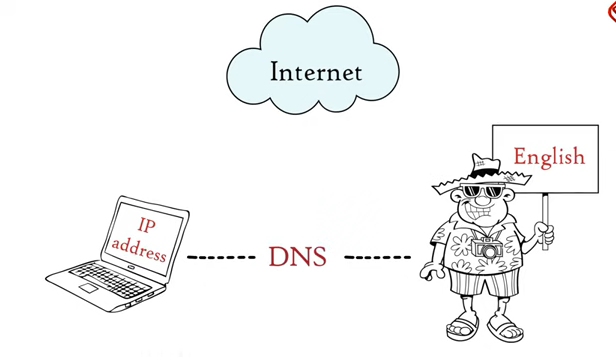
1. **Name space**
2. **DN space**
3. **Distribution of Name space**
4. **DNS in the internet**

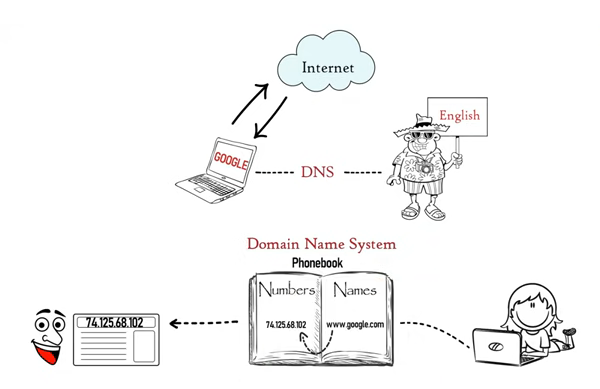
**DNS**

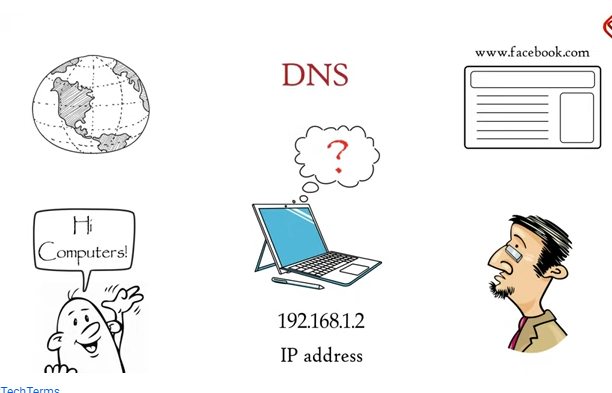
[**https://www.youtube.com/watch?v=l93k\_Lk6FyM&t=463s**](https://www.youtube.com/watch?v=l93k_Lk6FyM&t=463s)

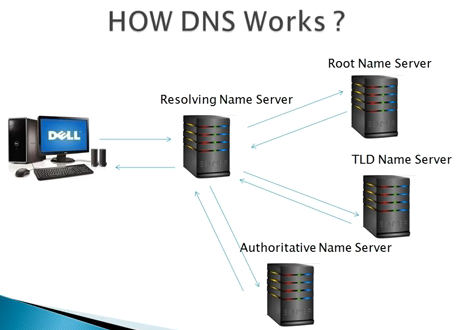
****

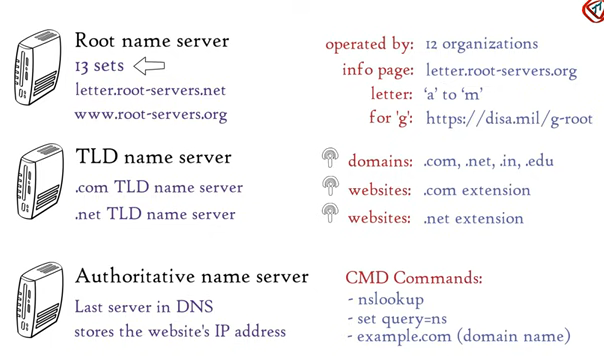




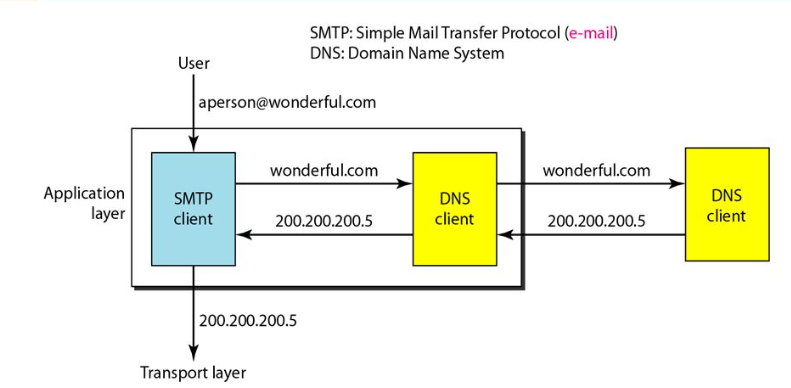








Example of how a DNS client/server program can support an e-mail program to find the IP address of an e-mail recipient:



* The Domain Name System (DNS) is a supporting program that is used by other programs such as e-mail.
* A user of an e-mail program may know the e-mail address of the recipient; however, the IP protocol needs the IP address.
* The DNS client program sends a request to a DNS server to map the e-mail address to the corresponding IP address.
* To identify an entity, TCP/IIP 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.

1. **Name Space**

* A name space that maps each address to a unique name can be organized in two ways:
  + **Flat Name space**
  + **Hierarchical Name space**

**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.**
* The names may or may not have a common section; if they do, it has no meaning.
* The **main disadvantage** of a fiat name space is that it cannot be used in a large system such as the Internet because it must be centrally controlled to avoid ambiguity and duplication.

**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.
* The management of the organization need not worry that the prefix chosen for a host is taken by another organization because, even if part of an address is the

Eg. Two colleges and one company want challenger in their address. But the central authority has given

College 1 – fhda.edu challenger.fhda.edu

College 2 – berkeley.edu challenger.berkely.edu

Company 1 - smart.com challenger.smart.com

1. **Domain Name Space**
   * To have a hierarchical name space, a domain name space was designed.
   * 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.

****

Domain name space

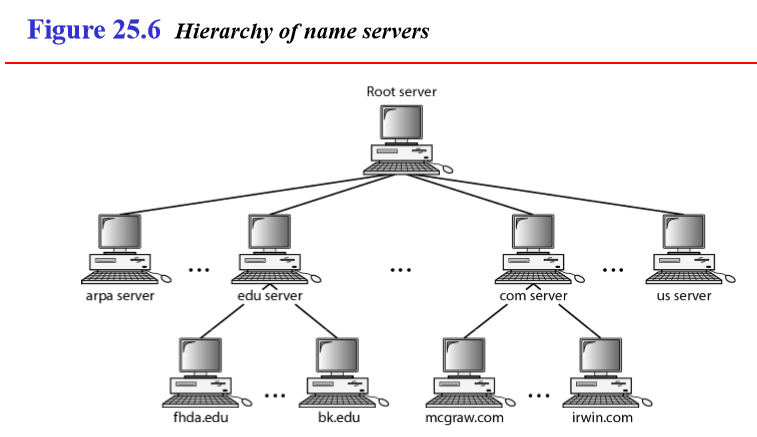
****

Domain names and labels

1. **Distribution of Name space**

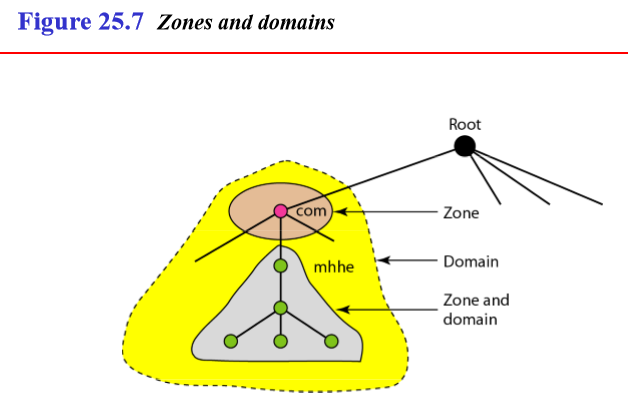
* The information contained in the domain namespace must be stored. However, it is very inefficient and also unreliable to have just one computer store such a huge amount of information.

1. Hierarchy of Name Servers
2. Zone
3. Root Server
4. Primary and Secondary Servers
5. Hierarchy of Name Servers

* The solution to these 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 (sub trees) 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 (sub domains).
* Each server can be responsible (authoritative) for either a large or a small domain.  
  

1. Zone

* Since the complete domain name hierarchy cannot be stored on a single server, it is divided among many servers. What a server is responsible for or has authority over is called a zone.
* The server makes a database called a zonefile and keeps all the information for every node under that domain.

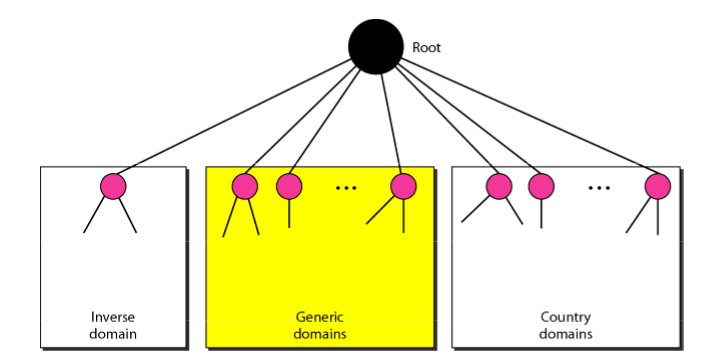


1. Root Server

* 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.

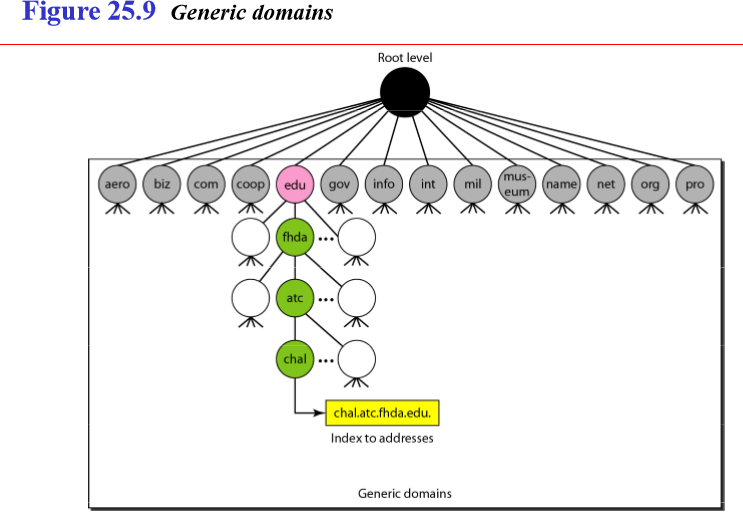
1. Primary and Secondary Servers
   * A primary server loads all information from the disk file;
   * the secondary server loads all information from the primary server.
   * When the secondary downloads information from the primary, it is called zone transfer.
2. **DNS in the internet**

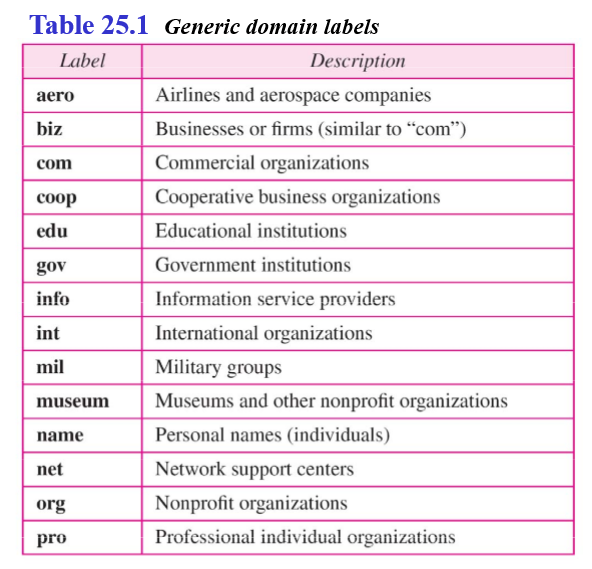
* DNS is a protocol that can be used in different platforms. In the Internet, the domain name space (tree) is divided into three different sections: generic domains, country domains, and the inverse domain



1. Generic domains

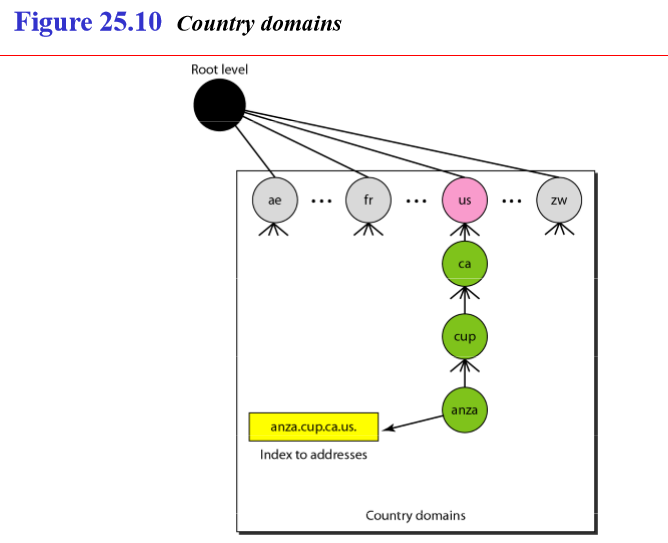
* The generic domains define registered hosts according to their generic behavior.





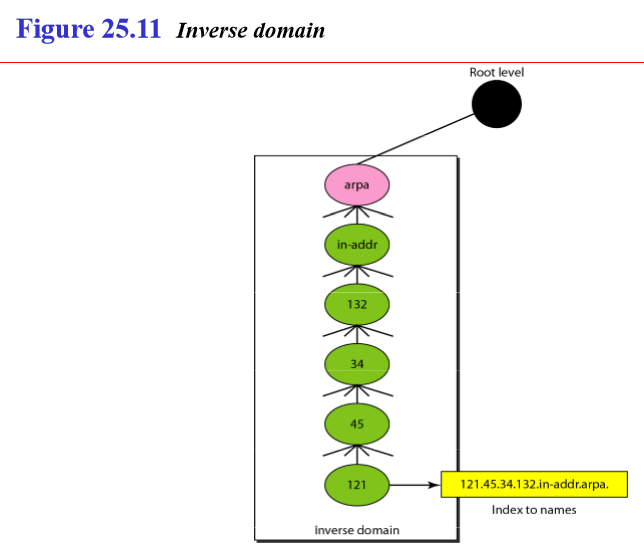
1. Country domains:

* 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.
* The United States, for example, uses state abbreviations as a subdivision of us (e.g., ca.us.).
* The address anza.cup.ca.us can be translated to De Anza College in Cupertino, California, in the United States.

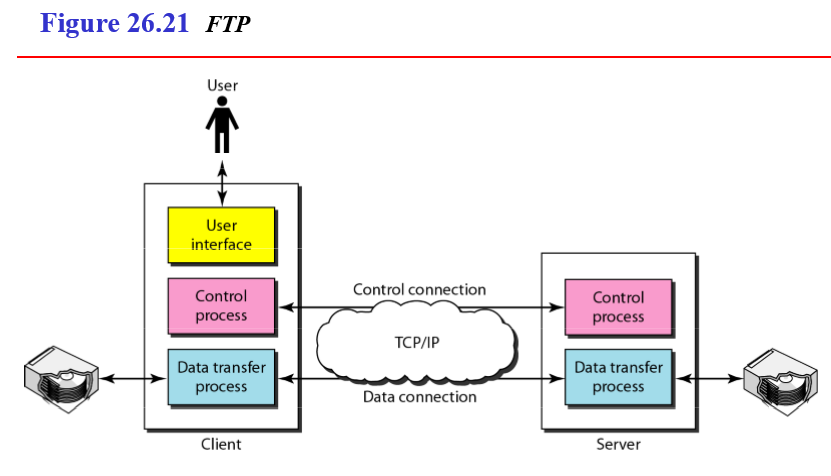


1. Inverse domain

* The inverse domain is used to map an address to a name.
* 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.



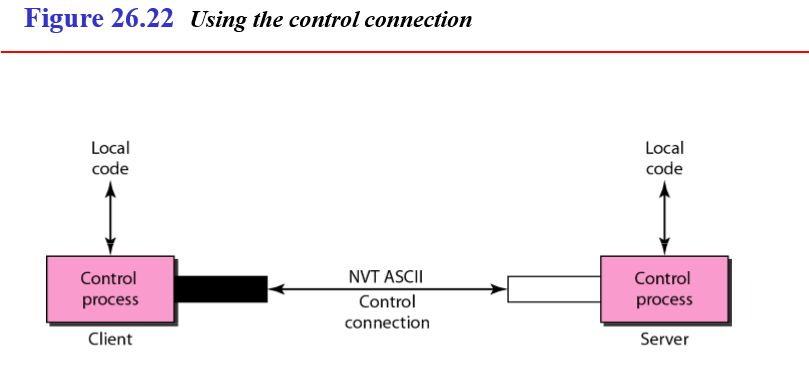
* 1. **File Transfer**
* File Transfer Protocol (FTP) is the standard mechanism provided by TCP/IP for copying a file from one host to another.
* FTP uses the services of TCP. It needs two TCP connections.
* The well-known port 21 is used for the control connection and the well-known port 20 for the data connection.



* The client has three components:
  + - user interface,
    - client control process
    - the client data transfer process.
* The server has two components:
  + - the server control process
    - the server data transfer process.
* The **control connection** is made between the **control processes.**
* The **data connection** is made between the **data transfer processes.**
* The control connection **remains connected during the entire interactive FTP session**. The data connection is **opened and then closed for each file transferred**. It opens each time commands that involve transferring files are used, and it closes when the file is transferred.
* **In other words, when a user starts an FTP session, the control connection opens. While the control connection is open, the data connection can be opened and closed multiple times if several files are transferred.**

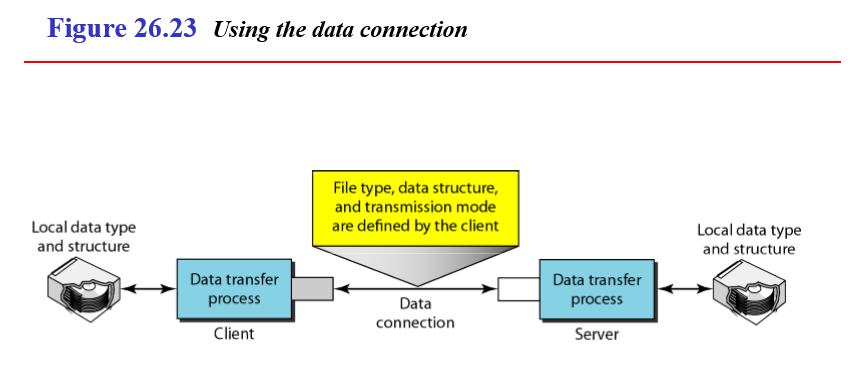
**Communication over Control Connection**

* It uses the 7-bit ASCII character set.
* Communication is achieved through commands and responses.
* This simple method is adequate for the control connection because we send one command (or response) at a time.
* Each command or response is only one short line, so we need not worry about file format or file structure.
* Each line is terminated with a two-character (carriage return and line feed) end-of-line token.

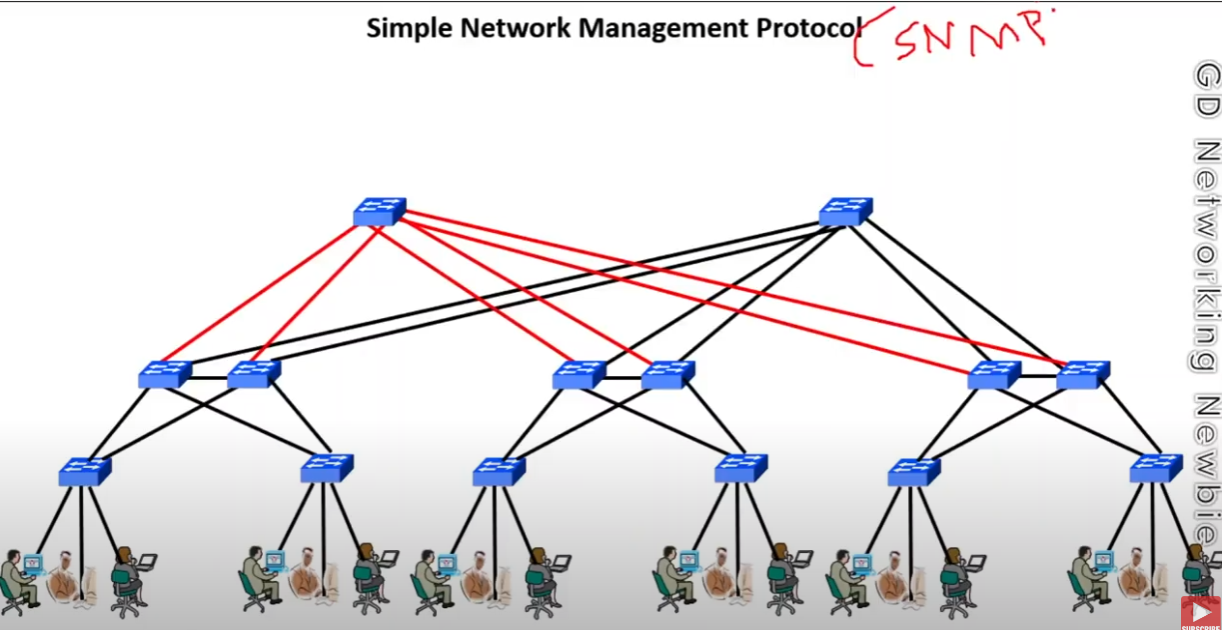


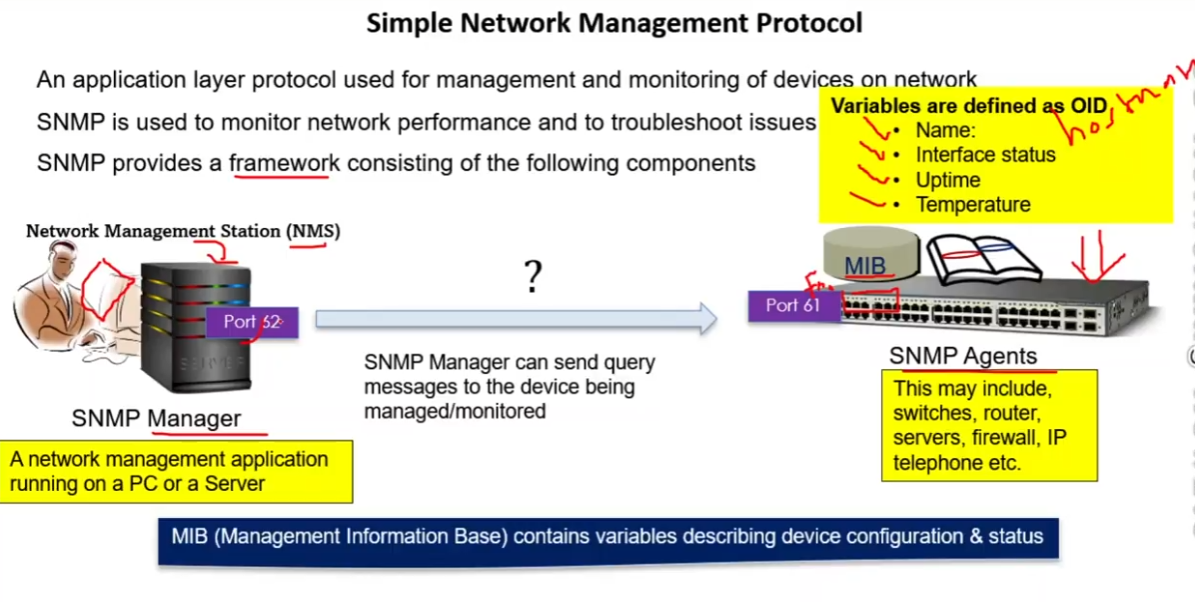
**Communication over Data Connection**

* we should remember that file transfer in FTP means one of three things:
  + A file is to be copied from the server to the client. This is called retrieving aft/e. It is done under the supervision of the RETR command,
  + A file is to be copied from the client to the server. This is called storing aft/e. It is done under the supervision of the STOR command.
  + A list of directory or file names is to be sent from the server to the client. This is done under the supervision of the LIST command. Note that FTP treats a list of directory or file names as a file. It is sent over the data connection.



* The client must define the type of file to be transferred, the structure of the data, and the transmission mode.
* Before sending the file through the data connection, we prepare for transmission through the control connection.
* The heterogeneity problem is resolved by defining **three attributes of communication: file type, data structure, and transmission mode**
  1. **Web Services**
  2. **SNMP**
  3. **HTTP**
  4. **E-Mail (SMTP,POP3,IMAP,MIME)**

****

****