PRACTICA II – REDES

**Capa de aplicación**

1. La capa de aplicación es responsable de soportar las aplicaciones de red. Ésta incluye muchos protocolos, entre ellos HTTP, SMTP y FTP entre otros. Un protocolo de la capa de aplicación es solo una parte de la aplicación de red.
2. a- Los dos procesos se comunican entre si enviando y recibiendo mensajes. Un proceso envia mensajes a la red y recibe mensaje de la red a través de su socket (puerta). El proceso emisor asume que existe una infraestructura de transporte al otro lado de la puerta que transportara el mensaje a través de internet hasta la puerta del proceso destino. Para que un proceso en un host pueda enviar mensajes a otro proceso de otro host es necesario que el proceso emisor identifique al proceso receptor, para identificarlo típicamente se deben especificar el nombre o dirección del host y un identificador que especifique al proceso receptor en el host de destino.

b- Si están en la misma maquina se comunican entre si utilizando comunicación interproceso, las reglas de la comunicación interproceso son gobernadas por el SO.

1. El modelo cliente-servidor consta de una parte cliente y otra servidor. Estos están en distintos sistemas terminales. Se dice cliente a aquel host que inicia la conexión y solicita un servicio al servidor. El host servidor, debe estar siempre encendido y tiene IP permanente, en cambio, el cliente por lo general se conecta intermitentemente y puede tener IP dinámica.

Un ejemplo de la vida cotidiana puede ser una estación de servicio, la que actua como servidor y nosotros los usuarios los clientes.

Un ejemplo informático puede ser un navegador web, el navegador implementa el lado cliente de HTTP y un servidor web implementa el lado servidor de HTTP.

Otro de los modelos de comunicación es la arquitectura P2P.

1. Un agente de usuario es una interfaz entre el usuario y la aplicación de red. Por ejemplo, considérese la web; el agente de usuario es un navegador, éste permite al usuario ver las paginas web, proporcionar entradas para formularios, etc. Por lo tanto el navegador es un proceso que envía/recibe mensajes por medio de un socket, y que proporciona una interfaz al usuario.

**DNS**

1. DNS (sistema de nombres de dominio) es una base de datos distribuida implementada en una jerarquía de servidores de nombres, y una aplicación de la capa de aplicación que permite que se comuniquen los host y los servicios de nombre para proporcionar el servicio de traducción. DNS es empleado comúnmente por otros protocolos de la capa de aplicación para traducir los nombres de host proporcionados por los usuarios a direcciones IP. El cliente DNS envia una consulta a un servidor DNS con el nombre del host, lo que constituye el mensaje DNS de peticion. El cliente DNS eventualmente recibe una respuesta que incluye la direccion IP para el nombre del host.
2. Los servidores DNS se organizan en forma jerárquica. Un host hace su solicitud a un **servidor local de nombres** que, si no posee la traducción, se transforma en un cliente consultando a un **servidor raíz de nombres -root server-**. Si el root server conoce al **servidor autorizado de nombres** para el dominio que se intenta resolver, envía la solicitud terminando la cadena; si lo desconoce, consulta a servidores intermedios hasta que eventualmente alguno lo conozca. En cada paso, un servidor podría tener registrado el IP del dominio consultado o mantenerlo en cache por una consulta anterior, ofreciendo una respuesta directa

Para un nombre de dominio, se denomina *Top Level Domain* (TLD) a la etiqueta más a la derecha, desde el último punto. Un **generic top-level domain** es una clasificación de los dominios de Internet, -un subconjunto de los TLD's- que actualmente incluye sólo las terminaciones: .com, .info, .net y .org. Otras clasificaciones mantienen por ejemplo códigos por país o administrados por distintas organizaciones.

1. Una respuesta authoritative es aquella entregada por el servidor oficial para el nombre de dominio consultado -es decir, que ningún servidor local, raíz ni intermedio respondió con información de caché-.
2. **- Consulta DNS Recursiva:** Un host realiza una consulta a un servidor, éste se encarga de obtener la resolución de IP y devuelve la respuesta al host.

**- Consulta DNS Iterativa:** Un name server realiza una consulta a un servidor; si este no tiene la respuesta le devuelve la IP del próximo servidor. El name server original se encarga de realizar nuevas solicitudes hasta obtener la respuesta.

1. Se denomina **resolver** a la parte cliente del DNS, encargada de iniciar y secuenciar las consultas. Usualmente trabajan de modo iterativo, aunque algunos sólo se conectan a un name server único y trabajan con consultas recursivas.
2. Los servidores de nombres almacenan registros de recursos (RR) para las recurrencias nombre de host/direccion IP. Cada mensaje de respuesta DNS transporta uno o mas registros de recursos. Un RR es una 4-tupla que contiene: (*Nombre, Tipo, Valor, TTL*). *TTL* es el tiempo de vida del registro de recurso; determina el momento en el que el recurso debe ser borrado de la cache.

El significado de *Nombre* y *Valor* dependen de *Tipo*:

* Tipo=A: Entonces *Nombre* es un nombre de host y *Valor* es la direccion IP del host. Por lo tanto, un registro de tipo A proporciona la correspondencia estandar de nombre de host a direcciones IP. Puede haber más de un registro tipo A con igual hostname y distintas IP en el caso de *servidores replicados* para realizar un *balance de carga*: la respuestas DNS incluirán la lista de asociaciones pero el servidor irá alternando las posiciones para distribuir los accesos de los clientes.
* Tipo=NS: Entonces *Nombre* es un dominio y *Valor* es el nombre de host de un servidor autorizado que sabe como obtener las direcciones IP de host en el dominio. Este registro se utiliza para rutar consultas DNS hacia adelante en la cadena de consultas.
* Si Tipo=CNAME: entonces *Valor* es un nombre de host canónico para el alias *Nombre*. Este registro puede proporcionar a los host que hacen consultas, el nombre canónico de un nombre de host.
* Si Tipo=MX: entonces *Valor* es el nombre canonico de un servidor de correo que tiene el alias nombre. Los registros MX permiten que los nombres de servidores de correo puedan tener alias sencillos.
* Si Tipo=PTR (Pointer Record) esExactamente al revés que un registro tipo A, indica en *Nombre* el IP de un host y en *Valor* su hostname. Permten la resolución reversa.
* Si Tipo=SOA.Por *Start of Authority*, indica que los registros siguientes de la base de datos corresponden a información autoritativa. Indica: nombre canónico del NS, usuario responsable -dirección de email con un punto en vez del arroba-, número de versión -serial- del archivo de configuración del NS, refresh o tiempo de espera de servidores secundarios para pedir el SOA al primario, expire o tiempo en segundos que tardarán los servidores secundarios en descartar los datos si no logran contactar al primario, y el mínimo valor TTL para los registros de recursos siguientes.

1. a) host [www.redes.unlp.edu.ar](http://www.redes.unlp.edu.ar/) : [www.redes.unlp.edu.ar](http://www.redes.unlp.edu.ar/) has adress 127.0.0.1

b) host -t NS redes.unlp.edu.ar : (para saber el nombre del servidor DNS autorizado) ns.redes.unlp.edu.ar

host ns.redes.unlp.edu.ar: 127.0.0.1

Todo en un solo comando: dig -t NS redes.unlp.edu.ar

c) host -t MX redes.unlp.edu.ar : redes.unlp.edu.ar mail is handled by 5 mail.redes.unlp.edu.ar.

host mail.redes.unlp.edu.ar has adress 127.0.0.1

O todo en un solo comando : dig -t MX redes.unlp.edu.ar

1. a) comando: host [www.google.com](http://www.google.com/)

respuesta: [www.google.com](http://www.google.com/) is an alias for [www.l.google.com](http://www.l.google.com/)

[www.l.google.com](http://www.l.google.com/) has adress 64.233.163.104

b)

c)

1. g
2. g

**HTTP**

1. Navegador: es un agente de usuario para la web, muestra la pagina web pedida y proporciona numerosas características de navegación y de configuración. Los navegadores web también implementan el lado cliente de HTTP.

Servidor web: este, alberga objetos web, cada uno de ellos direccionable por un URL. Los servidores web también implementan el lado servidor de HTTP.

Página web: también denominada documento, consta de objetos. Un objeto es simplemente un archivo, como por ejemplo un archivo HTML, una imagen jpg, o una imagen gif, etc. que es direccionable por un único URL. La mayoría de las paginas web estan formadas por un archivo HTML base y diversos objetos referenciados.

HTTP: define como los clientes web demandan paginas web, y cómo los servidores transfieren estas paginas web a los clientes. Es un protocolo de la capa de aplicación.

URL: es la dirección única de los objetos. Cada URL tiene dos componentes: el nombre del host del servidor que alberga el objeto, y el nombre de la ruta del objeto.

Cuando un usuario pide una pagina web, por ejemplo pulsa sobre un link, el navegador envía al servidor un mensaje HTTP de petición de los objetos de la pagina. El servidor recibe la petición y responde con un mensaje HTTP de respuesta que contiene los objetos.

1. Son dos cosas distintas, una (HTTP) es un protocolo, por lo tanto, define el formato y la forma de los mensajes que se envian entre si dos host que utilicen este protocolo. HTML es el lenguaje con el que se definen las paginas web, basicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondran una pagina web.
2. G
3. HTTP 1.0 utiliza conexiones no persistentes, es decir, solo se puede transferir un unico objeto web sobre una conexión. Se dice no persistente porque la conexión no persiste para los otros objetos.

HTTP 1.1 utiliza conexiónes persistenes en su modo por defecto, esto es, que no es necesario establecer una nueva conexión para la transferencia de cada objeto web, sino que se pueden transferir multiples objetos web durante el transcurso de una unica conexión.

1. H
2. h
3. HTTP utiliza el DNS.. SIEMPRE?
4. SMTP: *Simple Mail Transfer Protocol*, Transfiere mensajes desde el servidor de correo de un remitente al de un destinatario. Es mucho mas antiguo que HTTP.   
   Es importante observar que SMTP normalmente no utiliza servidores intermedios para enviar el correo.   
   POP: *Post Office Protocol*, Es un protocolo de acceso al correo extremadamente simple por ser tan sencillo su funcionalidad esta limitada. POP3 comienza cuando el cliente abre una conexión TCP con el servidor de correo sobre el puerto 110. En una transaccion POP3, el agente de usuario emite comandos, y el servidor responde a cada uno de ellos con una respuesta. Normalmente es configurado para borrar los emails tras bajarlos del servidor siendo imposible consultarlos desde otro equipo. Pero es posible configurarlos para *descargar* y *mantener* los mails.  
   IMAP: *Internet Mail Acess Protocol*, es un protocolo de acceso al correo, al igual que POP3. Tiene mucha mas funcionalidades que POP3 aunque es mas complejo. IMAP proporciona comandos que permiten a los usuarios crear buzones y mover los mensajes de un buzon a otro, IMAP tambien proporciona comandos para buscar en los buzones remotos mensajes que cumplen algun criterio en particular. Otra caracteristica importante es que tiene comandos que permiten al agente de usuario obtener componentes de mensajes.
5. Microsoft Exchange: servidor de comunicación basado en el correo electronico de colaboracion empresarial  
   Microsoft Outlook: cliente de correo electronico de Microsoft y puede ser utilizado como aplicación independiente o con Microsoft Exchange Server para dar servicios a multiples usuarios dentro de una organización.  
   Postfix: agente de transporte de correo (MTA) de software libre  
   Qmail: servidor de correo SMTP hecho para Unix, utiliza formado maildir para almacenar mensajes   
   Thunderbird: es un cliente de correo electronico de la fundacion Mozila, es software libre  
   Eudora: es un cliente de mail para SOs windows y Macintosh  
   Courier: agente de transporte de correo   
   Cyrus: agente de transporte de correo
6. sdfds