

Servidor de Correo Electrónico

Tema 2

Prof. Ing. Angel Brito Segura

1. Introducción

La suite de protocolos TCP/IP es la base de la comunicación en Internet y se considera el *backbone* de la mayoría de las redes que existen. En esta suite, se incluyen múltiples estándares que permiten la comunicación de datos, no importando el dispositivo que sea, quién lo fabricó, qué sistema operativo posee o donde este ubicado.

Los desarrolladores de protocolos de red siguen un enfoque en capas, donde cada capa es responsable de una porción diferente de la comunicación que se está produciendo en cualquier momento, como se observa en la figura 1:

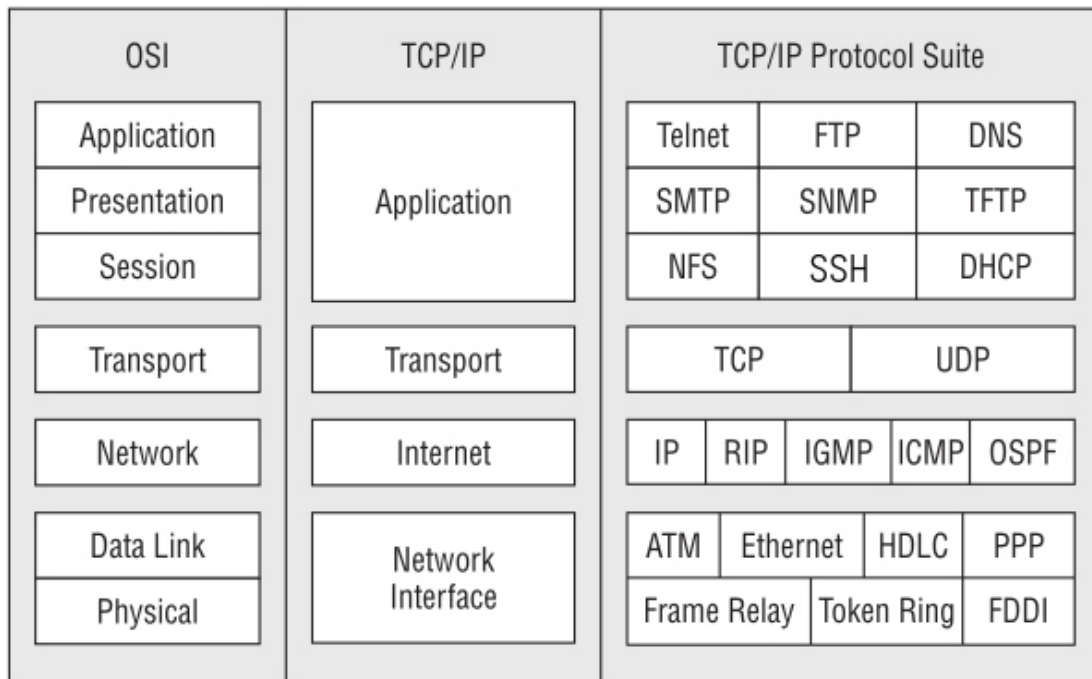


Figura 1: Comparación entre modelo OSI, TCP/IP y la suite de protocolos de Internet

La mayoría de las aplicaciones en la capa 4 del modelo TCP/IP utilizarán el método de comunicación cliente/servidor: donde uno de los nodos host actuará como servidor y el otro como cliente. Cada capa utilizará un protocolo o un grupo de ellos para transferir datos legibles desde la capa de origen a la capa destino como se observa en la siguiente figura 2 para el envío de un correo electrónico:

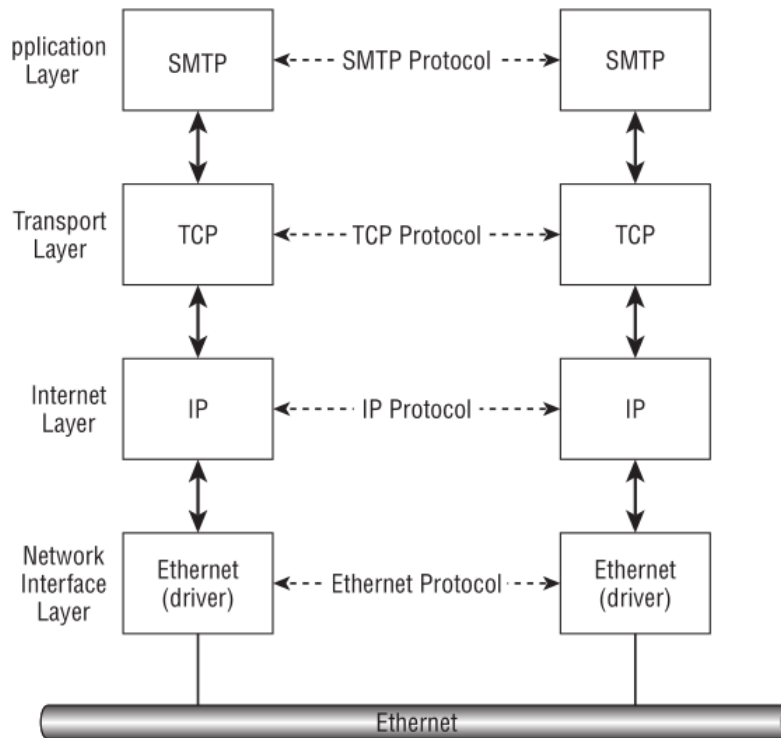


Figura 2: Capas de TCP/IP involucradas en el envío de un correo electrónico

2. Correo Electrónico

Considerado por muchos usuarios como el servicio de red más importante para las comunicaciones interpersonales, el correo electrónico es un servicio que permite enviar mensajes y archivos a través de una red de computadoras. Este servicio se puede dividir en dos diferentes tipos:

1. **Sistemas de correo electrónico basado en archivos:** Consisten en un conjunto de archivos guardados en una ubicación compartida (servidor). El servidor sólo proporciona acceso a los archivos, por lo que para tener conexión desde el exterior de éste generalmente se utiliza un *gateway server* (servidor de puerta de enlace) que maneja la interfaz del correo electrónico.
2. **Sistemas de correo electrónico cliente/servidor:** Consisten en un servidor que contiene los mensajes y maneja todas las interconexiones de correo electrónico, tanto dentro de la red local como con el exterior. Son más seguros y mucho más potentes que los basados en archivos y a menudo ofrecen funciones adicionales que permiten utilizar el sistema de correo electrónico para automatizar diferentes procesos empresariales.

2.1. Historia

Con más de 40 años de antigüedad, el correo electrónico ha evolucionado desde un simple sistema de mensajes de texto hasta un sistema de comunicación global que soporta mensajes de texto, imágenes, archivos adjuntos y mucho más.

Contrariamente a la creencia popular, el correo electrónico no llegó a la escena con el Internet, incluso es anterior a la Advanced Research Projects Agency Network (ARPANET, Red de la Agencia de Proyectos de Investigación Avanzada). El primer sistema de correo electrónico, un sistema de mensajería para múltiples usuarios en un sistema simple de tiempo compartido fue el MITs Compatible Time-Sharing System que se implementó en 1965.

El RFC 196 (Mail Box Protocol) de Dick Watson, publicado en julio de 1971 ya describía un sistema de correo electrónico. Mientras que **Ray Tomlinson** envió el primer mensaje de correo electrónico de red real a finales de año en la ARPANET, donde los mensajes se apoyaban en el Protocolo de Transferencia de Archivos (FTP).



Figura 3: Ray Tomlinson (1941-2016)

Como una forma asincrónica de interconectar de manera flexible computadoras que de otro modo no estarían conectadas, el Unix to Unix Copy Protocol (Protocolo de Copia Unix a Unix, UUCP, por sus siglas en inglés), fue creado en 1978 por los Laboratorios Bell. El UUCP permite el intercambio de datos entre computadoras sobre la base de una conectividad ad-hoc, como las conexiones telefónicas directas entre sistemas informáticos.

A diferencia de UUCP, y de manera mucho más parecida a Internet en la actualidad, la entonces predominante empresa Digital Equipment Corporation (pionera en la fabricación de minicomputadoras, adquirida por Compaq en 1998) ofrecía conectividad de red entre sus propias computadoras (DECnet) en 1975. Este servicio incluía funciones completas de intercambio y enrutamiento de mensajes.

De manera similar a UUCP, la “Because It’s Time Network” (BITNET) interconectaba unidades centrales de IBM en el ámbito académico y servicios similares de almacén e intercambio de mensajes iniciaron en 1981.

A principios de los años 80, la infraestructura de correo de Internet había sido creada, por lo que Postel y sus colegas realizaban esfuerzos para crear SMTP, el Protocolo de Transferencia de Correo Simple, con el IETF. En 1982, mientras ARPANET todavía estaba en transición del antiguo protocolo NCP (Network Control Protocol) al TCP/IP, Eric Allman desarrolló la primera implementación generalizada de correo

sobre SMTP con **sendmail**.

En ese momento, ya existían varias redes de intercambio de mensajes, para poder hacer frente a todas estas diferentes redes y sistemas de direcciones, se introdujo en **sendmail** un elaborado sistema de reescritura que le sirvió para manejar particularmente bien las diferencias en varios detalles como el direccionamiento entre SMTP, UUCP, BITNET y DECnet.

2.2. Arquitectura

En los comienzos del correo electrónico, el remitente y receptor eran usuarios (o programas de aplicaciones) en el mismo sistema; estaban conectados directamente a un sistema compartido, en el cual, sólo el administrador podía crear un buzón de correo donde se almacenaban los mensajes recibidos para cada usuario. Un buzón de correo (*mailbox*) era un archivo especial con restricciones de permisos dentro del disco duro local de cada máquina, donde sólo el propietario tiene acceso a él. Cada mensaje enviado contenía las direcciones de correo del remitente y destinatario (nombres de archivos).

Con la llegada de Internet, el remitente y receptor podían estar en diferentes sistemas, y por consiguiente, el mensaje necesitaba enviarse a través de esta red. El sistema (a veces llamado servidor de correo) en el sistema del remitente usaba una cola para almacenar los mensajes que esperan ser enviados, por su parte, el receptor también necesitaba recuperar los mensajes almacenados en el buzón de su sistema. Es por esto, que surge la necesidad de utilizar un cliente y un servidor que permita el envío del mensaje entre distintos sitios. Como la mayoría de los programas cliente/servidor en Internet, el servidor necesita estar en ejecución todo el tiempo porque no sabe cuándo un cliente solicitará una conexión. El cliente, por otro lado, puede ser alertado por el sistema cuando hay un mensaje en la cola para ser enviado.

Con esta revolución, era posible que el remitente estuviera apartado del servidor de correo, por lo que el mensaje podía ser enviado a través de una WAN punto a punto (módem de acceso telefónico), un DSL o módem de cable; o estar conectado a una LAN en una organización que utiliza un servidor de correo dedicado.

En la actualidad, tanto remitente como receptor pueden estar en cualquier lugar del mundo, y el mensaje puede ser enviado a través de Internet, LAN o WAN. Es necesario que el remitente tenga un programa para acceder a los mensajes (*pull program*) ya que para el envío de mensajes solo se han empujado a la cola (*push program*).

2.2.1. Agente de Usuario (UA)

Es un paquete de software (programa) que compone, lee, responde y reenvía mensajes de correo electrónico, así como gestiona buzones de correo. Los servicios típicos que brinda son:

- **Composición de mensajes:** Proporcionan plantillas para ayudar al usuario a redactar el mensaje que va a enviar. Algunos incluso tienen un editor integrado que puede realizar la corrección ortográfica y gramatical y otras tareas que se esperan de un procesador de textos sofisticado. Por supuesto, el usuario también puede utilizar su propio editor de texto para crear el mensaje e importarlo, o cortarlo y pegarlo, en la plantilla del agente de usuario.

- **Lectura de mensajes:** Cuando un usuario lo invoca, primero comprueba el correo en el buzón de entrada. La mayoría de los agentes de usuario muestran un resumen de una línea de cada correo recibido. Cada correo electrónico contiene los siguientes campos:
 1. Un campo de número.
 2. Una bandera que muestra el estado del correo, como nuevo, ya leído pero no respondido o leído y respondido.
 3. El tamaño del mensaje.
 4. El remitente.
 5. El campo de asunto (opcional).
- **Respuesta a mensajes:** Generalmente permite al usuario responder al remitente original o responder a todos los destinatarios del mensaje. El mensaje de respuesta puede contener el mensaje original (para una referencia rápida) y el mensaje nuevo.
- **Reenvío de mensajes:** Permite al receptor tanto enviar un mensaje al remitente o a los destinatarios de la copia, como reenviar (enviar el mensaje a un tercero) reenviar el mensaje, con o sin comentarios adicionales.
- **Gestión de buzones de correo:** Normalmente se crean dos buzones de correo: una bandeja de entrada y una de salida. Cada buzón es un archivo con un formato especial que puede ser manejado por el agente de usuario. La bandeja de entrada guarda todos los correos electrónicos recibidos hasta que el usuario los elimine. La bandeja de salida guarda todos los correos electrónicos enviados hasta que el usuario los elimine. La mayoría de los agentes de usuario actuales son capaces de crear buzones de correo personalizados.

Existen dos tipos de agentes de usuario:

1. **Basados en comandos:** Son los primeros en aparecer y todavía están presentes como agentes de usuario subyacentes en los servidores. Un agente de usuario basado en comandos normalmente acepta un comando de un carácter del teclado para realizar su tarea. Por ejemplo, un usuario puede escribir el carácter `r` en el símbolo del sistema para responder al remitente del mensaje, o escribir el carácter `R` para responder al remitente y a todos los destinatarios. Algunos ejemplos de agentes de usuario basados en comandos son `mail`, `pine` y `elm`.
2. **Basados en GUI:** Contienen componentes de interfaz gráfica de usuario (GUI) que permiten al usuario interactuar con el software utilizando tanto el teclado como el ratón. Tienen componentes gráficos como iconos, barras de menú y ventanas que facilitan su uso. Algunos ejemplos de agentes de usuario basados en GUI son *Outlook*, *Thunderbird* y *Eudora*.

2.2.2. Agente de Transferencia de Correo (MTA)

En sí, es el servidor de correo electrónico donde se almacenan y reenvían los mensajes del correo electrónico. El MTA es responsable de recibir mensajes de correo electrónico de los agentes de usuario y de reenviarlos a los servidores de correo electrónico de destino.

2.2.3. Agente de Acceso al Correo (MAA)

Es el sistema que se encarga del acceso al correo almacenado, haciendo accesible los buzones en cualquier equipo del mundo. El MAA es responsable de la autenticación del usuario y de la entrega de mensajes de correo electrónico a los agentes de usuario.

2.3. Envío de correo electrónico

Para realizar esta acción, el usuario (a través del UA), crea un correo electrónico que se observa muy similar a los correos postales. Esto se conforma de:

- **Sobre (envelope):** Generalmente contiene las direcciones del remitente y del destinatario.
- **Mensaje:** Contiene encabezado y cuerpo: el *encabezado* del mensaje define el remitente, receptor, asunto del mensaje y alguna otra información (como el tipo de codificación). El cuerpo del mensaje contiene la información real que el destinatario debe leer.

2.4. Recepción de correos electrónicos

El UA es activado por el usuario (o por un temporizador) e informa con una notificación al usuario si cuenta con correo electrónicos. Si el usuario está listo para leer, se muestra un archivo `mail.alist` en el que cada línea contiene un resumen de la información sobre un mensaje en particular en el buzón. Este resumen generalmente incluye la dirección de correo del remitente, el asunto y la hora en que se envió o recibió el correo. El usuario puede seleccionar cualquiera de los mensajes y mostrar su contenido en la pantalla.

2.5. Direcciones

Para entregar los correos electrónicos, un sistema de gestión de correo debe utilizar un sistema de direcciones con direccionamiento único. En Internet, la dirección de correo electrónico consta de dos partes:

1. **Nombre de usuario** (parte local): Define el nombre del buzón de usuario donde se almacena todo el correo recibido por un usuario para que lo recupere el MAA.
2. **Nombre de dominio:** Una organización generalmente selecciona uno o más hosts para recibir y enviar correo electrónico; los hosts a veces se denominan servidores de correo o *exchangers*. El nombre de dominio asignado a cada intercambiador de correo o bien proviene de la base de datos DNS o es un nombre lógico (por ejemplo, el nombre de la organización).

2.6. Lista de correo

Habilidad del correo electrónico que permite que un alias, represente varias direcciones de correo electrónico diferentes. Cada vez que se envía un mensaje, el sistema comprueba el nombre del destinatario con la base de datos de alias; si existe una lista de correo para el alias definido, se deben preparar mensajes separados, uno para cada entrada de la lista, y entregarlos al MTA. Si no existe una lista de correo para el alias, el nombre en sí es la dirección de recepción y se entrega un solo mensaje a la entidad de transferencia de correo.

2.7. MIME

El correo electrónico tiene una estructura sencilla por lo que sólo se podían enviar mensajes en formato ASCII de 7 bits NVT (Network Vulnerability Test). Además, no se podía utilizar para enviar archivos binarios o datos de vídeo/audio. El protocolo de correo electrónico originalmente solo podía manejar texto sin formato, pero con el tiempo, se desarrolló el protocolo Multipurpose Internet Mail Extensions (MIME) para permitir que los mensajes de correo electrónico contengan datos binarios, como imágenes, archivos de audio y video, y otros tipos de datos no textuales. MIME también permite a los usuarios enviar mensajes de correo electrónico en varios formatos de texto, como HTML, RTF y otros formatos de texto enriquecido.

MIME (Multipurpose Internet Mail Extensions) es un protocolo complementario que permite enviar datos no ASCII a través del correo electrónico. Este protocolo transforma los datos no ASCII en el sitio del remitente en datos ASCII NVT para entregarlos al MTA del cliente y se envían a través de Internet. Una vez que se recibe el mensaje, se transforma de nuevo en los datos originales para que el receptor lo pueda leer sin problemas.

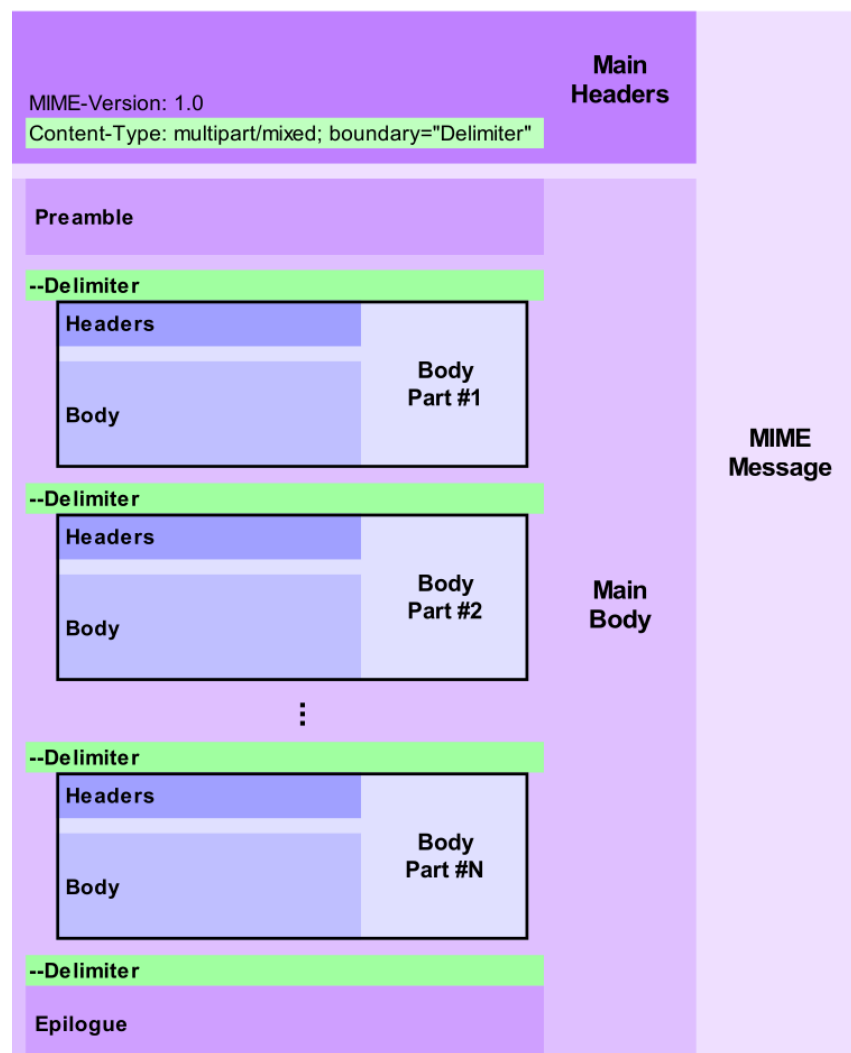


Figura 4: Estructura de mensajes multiparte MIME

Este protocolo define cinco encabezados que se pueden agregar al correo electrónico original para definir los parámetros de transformación:

2.7.1. MIME-Version

Este encabezado define la versión de MIME a utilizar.

2.7.2. Content-Type

Este encabezado define el tipo de dato que se utiliza en el cuerpo del mensaje. El tipo de contenido y el subtipo están separados por una barra (según el subtipo, el encabezado puede contener otros parámetros) y permite siete diferentes tipos de datos como se muestra en la figura 5:

<i>Type</i>	<i>Subtype</i>	<i>Description</i>
Text	Plain	Unformatted
	HTML	HTML format
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to mixed subtypes, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	IPEG	Image is in IPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single-channel encoding of voice at 8 kHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (8-bit bytes)

Figura 5: Tipos y subtipos de datos en MIME

2.7.3. Content-Transfer-Encoding

Este encabezado define el método utilizado para codificar los mensajes en ceros y unos para su transporte. Cuenta con cinco tipos de métodos de codificación que se enumeran en la figura 5:

<i>Type</i>	<i>Description</i>
7-bit	NVT ASCII characters and short lines
8-bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base-64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equals sign followed by an ASCII code

Figura 6: Codificación del contenido en la transferencia

2.7.4. Content-Id

Este encabezado identifica de forma única todo el mensaje en un entorno de múltiples mensajes.

2.7.5. Content-Description

Este encabezado proporciona una descripción del contenido del mensaje, definiendo si el cuerpo es imagen, audio o video.

La suite de protocolos TCP/IP proporciona un sistema de correo electrónico confiable y flexible construido con pocos protocolos básicos.

3. Simple Mail Transfer Protocol (SMTP)

La mayoría de las aplicaciones estándar TCP/IP se les asigna un puerto conocido para que los sistemas remotos sepan cómo conectar el servicio. SMTP es el protocolo base de todos los sistemas de correo electrónico sobre TCP/IP.

La transferencia de correo propiamente dicha se realiza a través de los agentes de transferencia de mensajes (MTAs). Para enviar correo, un sistema debe tener el MTA cliente y, para recibir correo, un sistema debe tener un MTA servidor. El protocolo formal que define el cliente y el servidor MTA en Internet se denomina Protocolo Simple de Transferencia de Correo electrónico (SMTP).

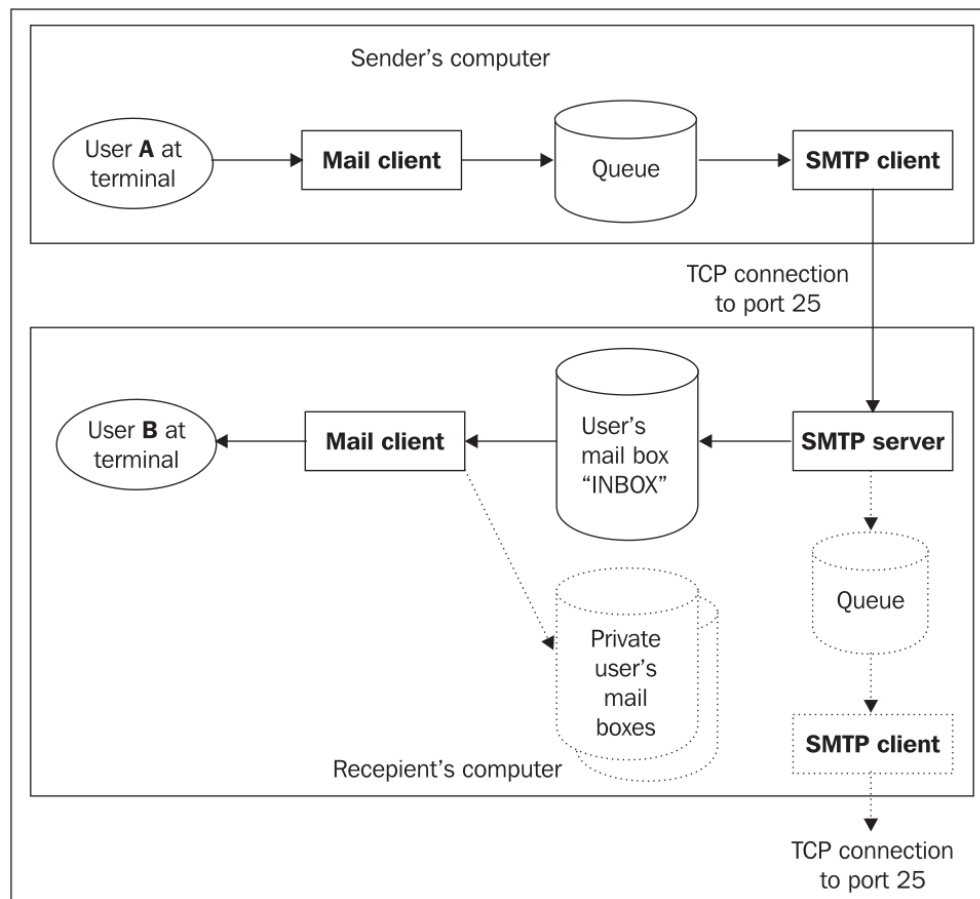


Figura 7: Arquitectura SMTP

Como su nombre lo indica, el protocolo Simple de es utilizado para transportar correo a través de Internet o de la red local. El protocolo SMTP fue definido originalmente en el RFC 821, es un servicio confiable y orientado a la conexión que proporciona el TCP y se indica que el puerto de escucha será el número 25.

Actualmente, el protocolo sigue el [RFC 2821](#) donde se utiliza dos veces: entre el remitente y su servidor de correo y entre los dos servidores de correo. Este protocolo simplemente define cómo deben enviarse y recibirse los comandos y las respuestas del correo electrónico. Cada red es libre de elegir un paquete de software para su implementación.

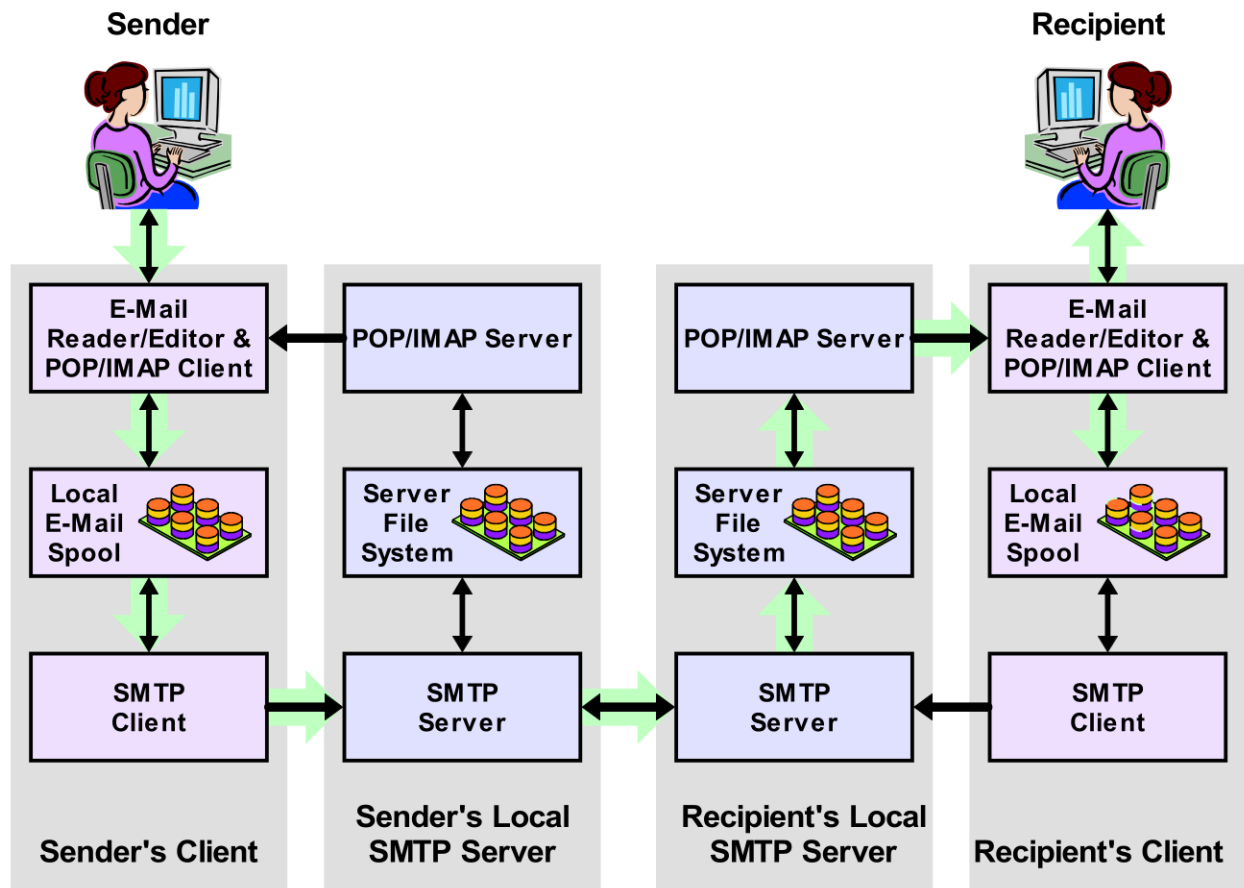


Figura 8: Modelo de comunicación del correo electrónico

SMTP utiliza comandos y respuestas para transferir mensajes entre un cliente MTA y un servidor MTA como se muestra en la figura 9:

Command	Syntax	Function
Hello	HELO <sending-host> EHLO <sending-host>	Identify sending SMTP
From	MAIL FROM:<from-address>	Sender address
Recipient	RCPT TO:<to-address>	Recipient address
Data	DATA	Begin a message
Reset	RSET	Abort a message
Verify	VERFY <string>	Verify a username
Expand	EXPN <string>	Expand a mailing list
Help	HELP [string]	Request online help
Quit	QUIT	End the SMTP session

Figura 9: Comandos comunes SMTP

<i>Code</i>	<i>Description</i>
Positive Completion Reply	
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
Positive Intermediate Reply	
354	Start mail input
Transient Negative Completion Reply	
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted: insufficient storage
Permanent Negative Completion Reply	
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

Figura 10: Respuestas SMTP

SMTP proporciona una entrega de correo electrónico directa de extremo a extremo. La entrega directa permite que siempre se entregue el correo electrónico sin depender de hosts intermedios. Si la entrega falla, el sistema local lo sabe de inmediato, por lo que informa al usuario que envió el correo o ponerlo en cola para una entrega posterior sin depender de sistemas remotos.

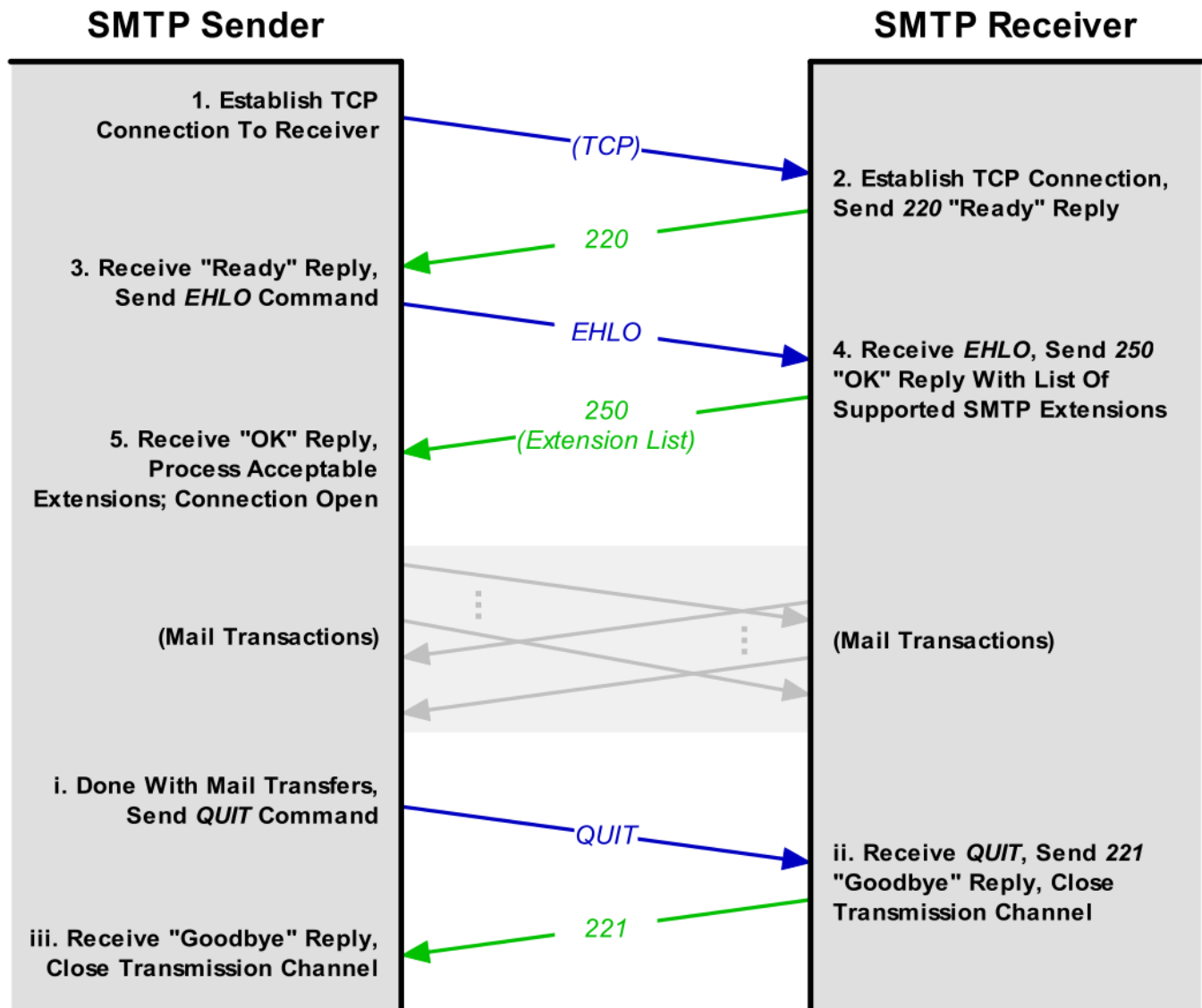


Figura 11: Administración sesión del SMTP

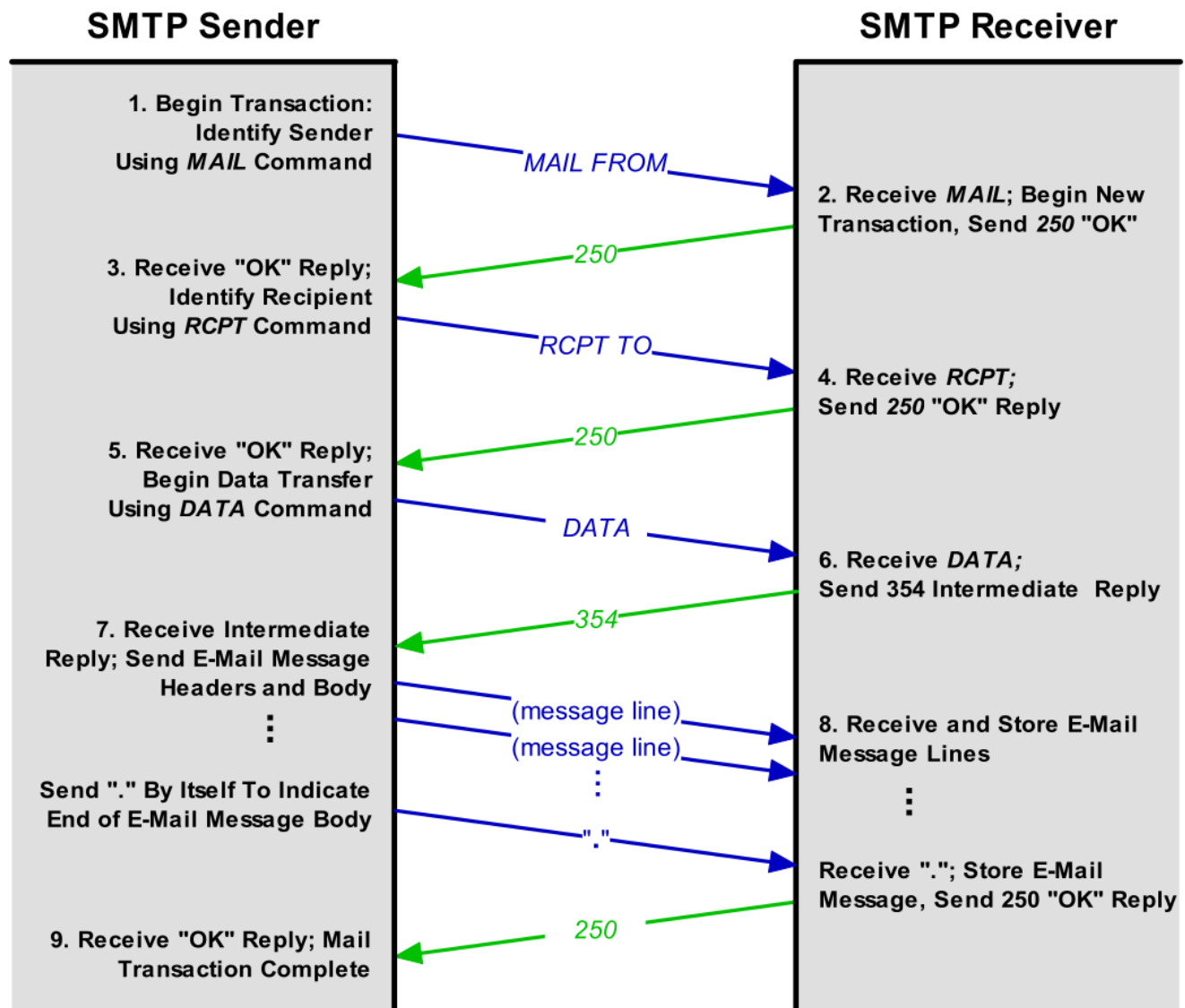


Figura 12: Proceso de transacción de correo SMTP

La desventaja de la entrega directa es que requiere que ambos sistemas sean totalmente capaces de manejar correo. Algunos de estos sistemas suelen apagarse al final del día y con frecuencia se encuentran fuera de línea, por lo que se utilizan funciones de un sistema DNS para enrutar el mensaje a un servidor de correo en lugar de la entrega directa. Luego, el correo se traslada desde el servidor al sistema cliente cuando el cliente vuelve a estar en línea. Uno de los protocolos que utilizan las redes TCP/IP para esta tarea es POP.

4. Post Office Protocol (POP)

El Protocolo de Oficina Postal (POP, por sus siglas en inglés) es un protocolo simple basado en texto utilizado por los clientes de correo electrónico para recuperar mensajes de un servidor de correo remoto. Existen dos versiones de este protocolo, las cuales son incompatibles ya que utilizan comandos diferentes, aunque realizan las mismas funciones básicas limitadas. La versión 2 (POP2) ya no es utilizada con tanta

frecuencia, fue definida en el RFC 937 y utiliza el puerto 109. Mientras que su versión posterior (**POP3**) es definido en el RFC 1939, utiliza el puerto 110 y es un protocolo de solicitud/respuesta simple. Cualquier protocolo POP verifica el nombre de usuario y su contraseña, una vez autenticado, traslada el correo electrónico del usuario que se encuentra en un servidor SMTP al lector local de correo electrónico que está utilizando.

4.1. RFC 1939

Esta solicitud de comentarios define el Post Office Protocol en su versión 3 (POP3) definiendo que la autenticación se debe proporcionar desde el cliente con un usuario y contraseña para acceder a la bandeja de entrada. También nos indica que puede funcionar en dos modos principales:

- **Modo de eliminación:** Como su nombre sugiere, el correo electrónico es eliminado del buzón del servidor SMTP después de cada recuperación. Este modo se utiliza normalmente cuando el usuario está trabajando en su computadora permanente y puede guardar y organizar el correo recibido después de leerlo o responderlo.
- **Modo de conservación:** como su nombre indica, el correo electrónico permanece en el buzón del servidor SMTP después de la recuperación. Este modo, es utilizado normalmente cuando el usuario accede a su correo fuera de su computadora principal, donde el correo sólo se lee, pero se conserva en el servidor para recuperarlo y organizarlo más tarde.

Además, nos indica los comandos mostrados en la siguiente figura 13 que se pueden utilizar dentro de este protocolo:

Command	Function
USER <i>username</i>	The user's account name
PASS <i>password</i>	The user's password
STAT	Display the number of unread messages/bytes
RETR <i>n</i>	Retrieve message number <i>n</i>
DELE <i>n</i>	Delete message number <i>n</i>
LAST	Display the number of the last message accessed
LIST [<i>n</i>]	Display the size of message <i>n</i> or of all messages
RSET	Undelete all messages; reset message number to 1
TOP <i>n</i> /	Print the headers and / lines of message <i>n</i>
NOOP	Do nothing
QUIT	End the POP3 session

Figura 13: Comandos POP3

Para los grandes proveedores de Internet, el protocolo POP3 es muy ventajoso porque el correo electrónico de sus usuarios no se queda en el servidor SMTP que manejan:

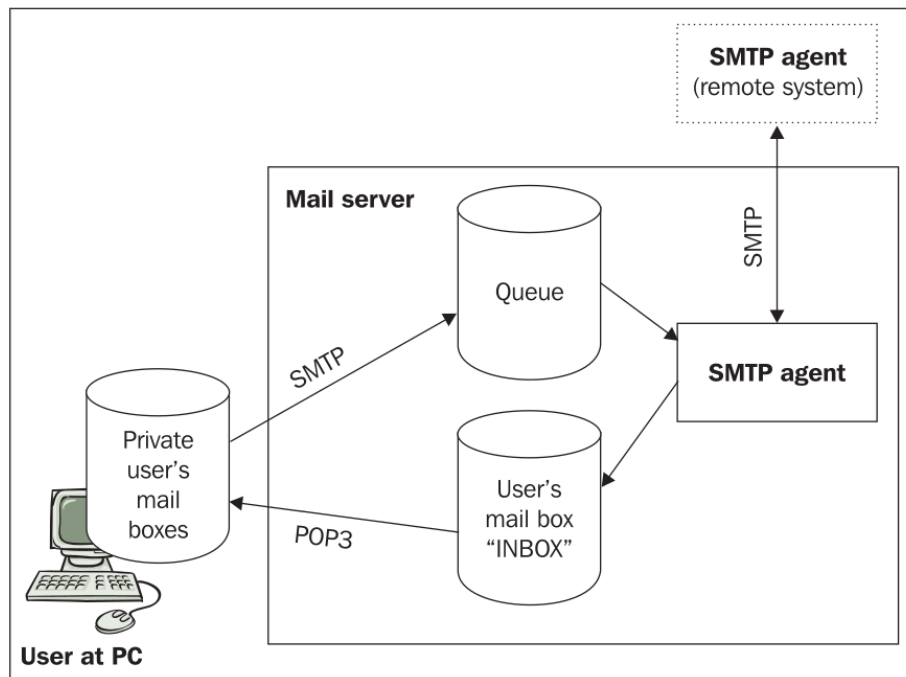


Figura 14: POP3 en proveedores de Internet

4.2. Proceso general

En la siguiente máquina de estados finita se muestra el proceso general dentro del protocolo POP3:

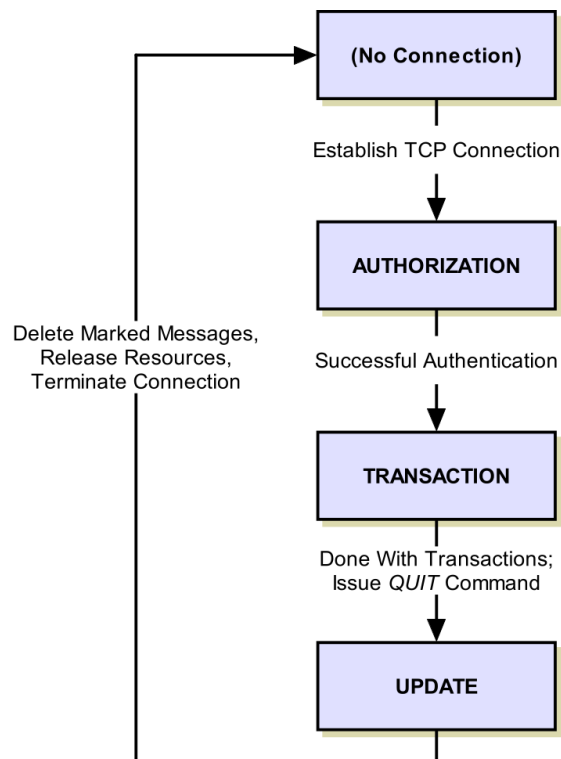


Figura 15: Máquina de estados finita del protocolo POP3

Para realizar la autenticación el comando USER proporciona el nombre de usuario, mientras que el comando PASS la contraseña para la cuenta del buzón que se está recuperando (mismo que se utiliza para iniciar sesión en el servidor de correo) como se muestra en la siguiente figura 16:

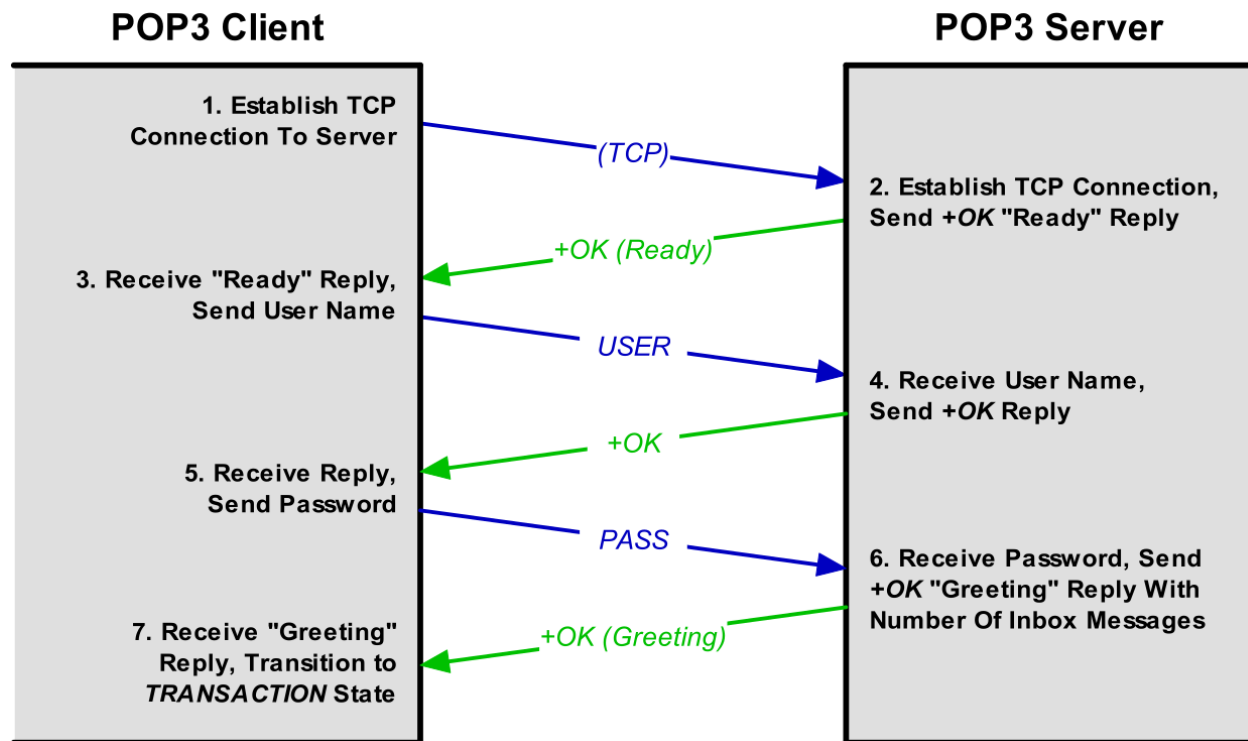


Figura 16: Proceso de Autenticación de Usuarios en POP3

Una vez autenticado, se envía el comando STAT, el servidor le responde con un recuento de la cantidad de mensajes en el buzón y la cantidad total de bytes contenidos en esos mensajes. RETR 1 recupera el texto completo del primer mensaje, mientras que DELE 1 elimina ese mensaje del servidor. Cada mensaje se recupera y se elimina a su vez como se observa en la figura 17. El cliente finaliza la sesión con el comando QUIT.

Los comandos de recuperación RETR y eliminación DELE utilizan números de mensaje que permiten que los correos electrónicos se procesen en cualquier orden. Además, no existe un vínculo directo entre recuperar un mensaje y eliminarlo. Es posible eliminar un mensaje que nunca se ha leído o conservar un mensaje incluso después de haberlo leído.

En un servidor POP promedio, todo el contenido del buzón se traslada al cliente y se elimina del servidor o se conserva como si nunca se hubiera leído. La eliminación de mensajes individuales en el cliente no se refleja en el servidor porque todos los mensajes se tratan como una sola unidad que se elimina o se conserva después de la transferencia inicial de datos al cliente. Los clientes de correo electrónico que desean mantener de forma remota un buzón de correo electrónico deben de utilizar el protocolo IMAP.

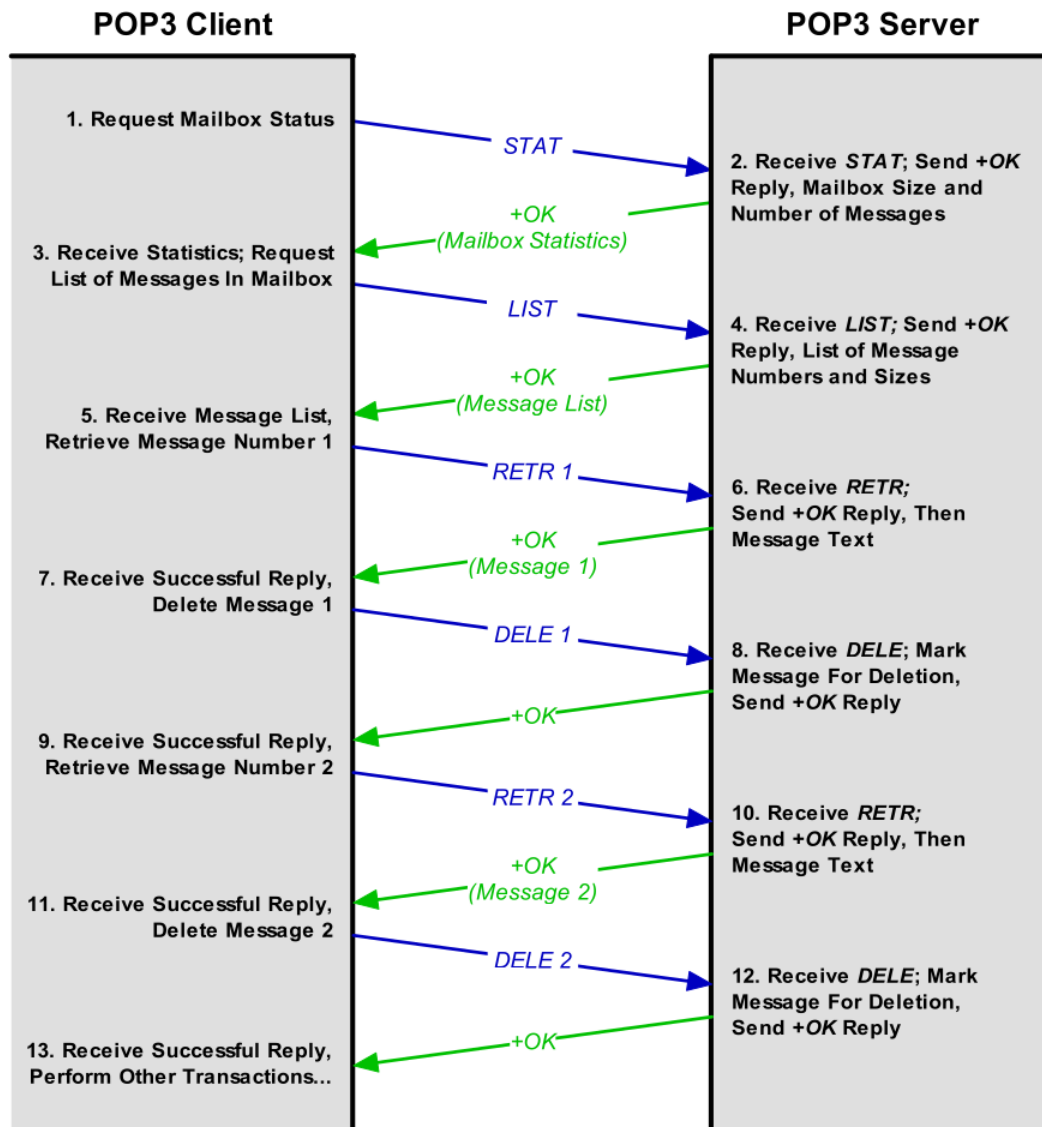


Figura 17: Proceso de recuperación de mensajes (*mail exchange*) en POP3

5. Internet Message Access Protocol (IMAP)

El Protocolo de Acceso a Mensajes de Internet (IMAP) es una alternativa a POP ya que proporciona el mismo servicio básico que POP y añade funciones para admitir la sincronización de buzones (capacidad de leer mensajes de correo individuales en un cliente o directamente en el servidor mientras se mantienen los buzones de correo en ambos sistemas completamente actualizados), utiliza el puerto 143 para la comunicación TCP y es un protocolo de solicitud/respuesta con un pequeño conjunto de comandos.

5.1. RFC 3501

Esta solicitud de comentarios define el Internet Message Access Protocol versión 4rev1 (IMAP4rev1), protocolo de acceso a correos electrónicos en servidores remotos que permite la administración avanzada

del correo directamente en el servidor, facilitando el acceso desde múltiples dispositivos sin necesidad de descargar los mensajes localmente.

El conjunto de comandos de este protocolo es algo más complejo que el utilizado por POP y, como se muestra en la siguiente figura 18, está claramente orientado hacia los buzones de correo para mantenerlos de forma remota lo que se almacena en el servidor SMTP.

Command	Function
CAPABILITY	List the features supported by the server
NOOP	Literally "No Operation"
LOGOUT	Close the connection
AUTHENTICATE	Request an alternate authentication method
LOGIN	Provide the username and password for plain-text authentication
SELECT	Open a mailbox
EXAMINE	Open a mailbox as read-only
CREATE	Create a new mailbox
DELETE	Remove a mailbox
RENAME	Change the name of a mailbox
SUBSCRIBE	Add a mailbox to the list of active mailboxes
UNSUBSCRIBE	Delete a mailbox name from the list of active mailboxes
LIST	Display the requested mailbox names from the set of all mailbox names
LSUB	Display the requested mailbox names from the set of active mailboxes
STATUS	Request the status of a mailbox
APPEND	Add a message to the end of the specified mailbox
CHECK	Force a checkpoint of the current mailbox
CLOSE	Close the mailbox and remove all messages marked for deletion
EXPUNGE	Remove from the current mailbox all messages marked for deletion
SEARCH	Display all messages in the mailbox that match the specified search criterion
FETCH	Retrieve a message from the mailbox
STORE	Modify a message in the mailbox
COPY	Copy the specified messages to the end of the specified mailbox
UID	Locate a message based on the message's unique identifier

Figura 18: Comandos IMAP4

5.2. Proceso general

En la siguiente máquina de estados finita se muestra el proceso general dentro del protocolo IMAP4:

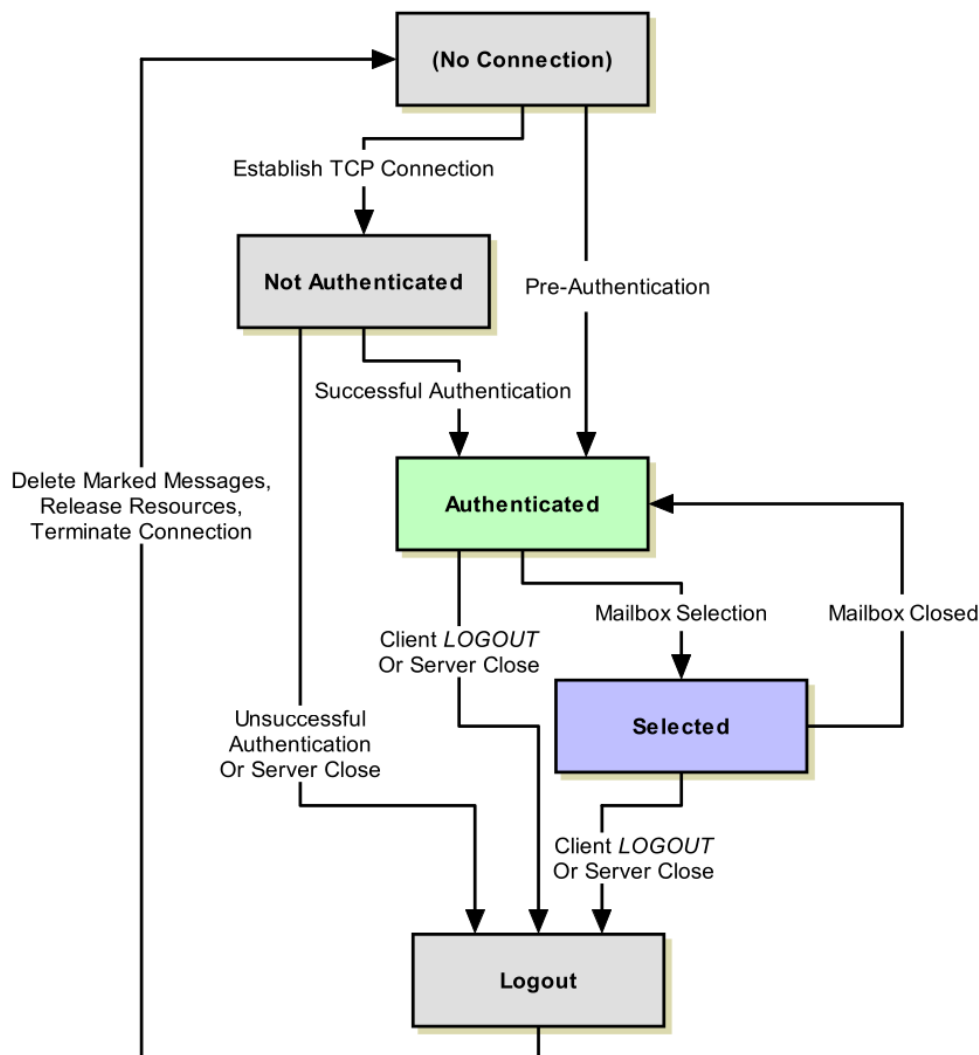


Figura 19: Máquina de estados finita del protocolo IMAP4

El primer comando IMAP ingresado por el usuario es LOGIN, que proporciona el nombre de usuario y la contraseña, el comando está precedido por una etiqueta, que es un identificador único generado por el cliente para cada comando. El comando SELECT permite seleccionar que buzón de correo se utilizará. El servidor IMAP muestra el estado del buzón de correo. Asociados con cada mensaje hay una serie de indicadores que se utilizan para administrar los mensajes en el buzón de correo marcándolos como Visto, No visto, Eliminado, etc.

El comando FETCH descarga un mensaje del buzón tanto el texto como sólo los encabezados o indicadores. Una vez descargado el mensaje, el usuario lo elimina. Esto se hace escribiendo el indicador *Deleted* con el comando STORE. El comando DELETE no se utiliza para eliminar mensajes; elimina buzones completos.

Los mensajes individuales se marcan para su eliminación estableciendo el indicador *Delete* y no se eliminan hasta que se emite el comando EXPUNGE o se cierra explícitamente el buzón con el comando CLOSE. Luego, la sesión se termina con el comando LOGOUT.

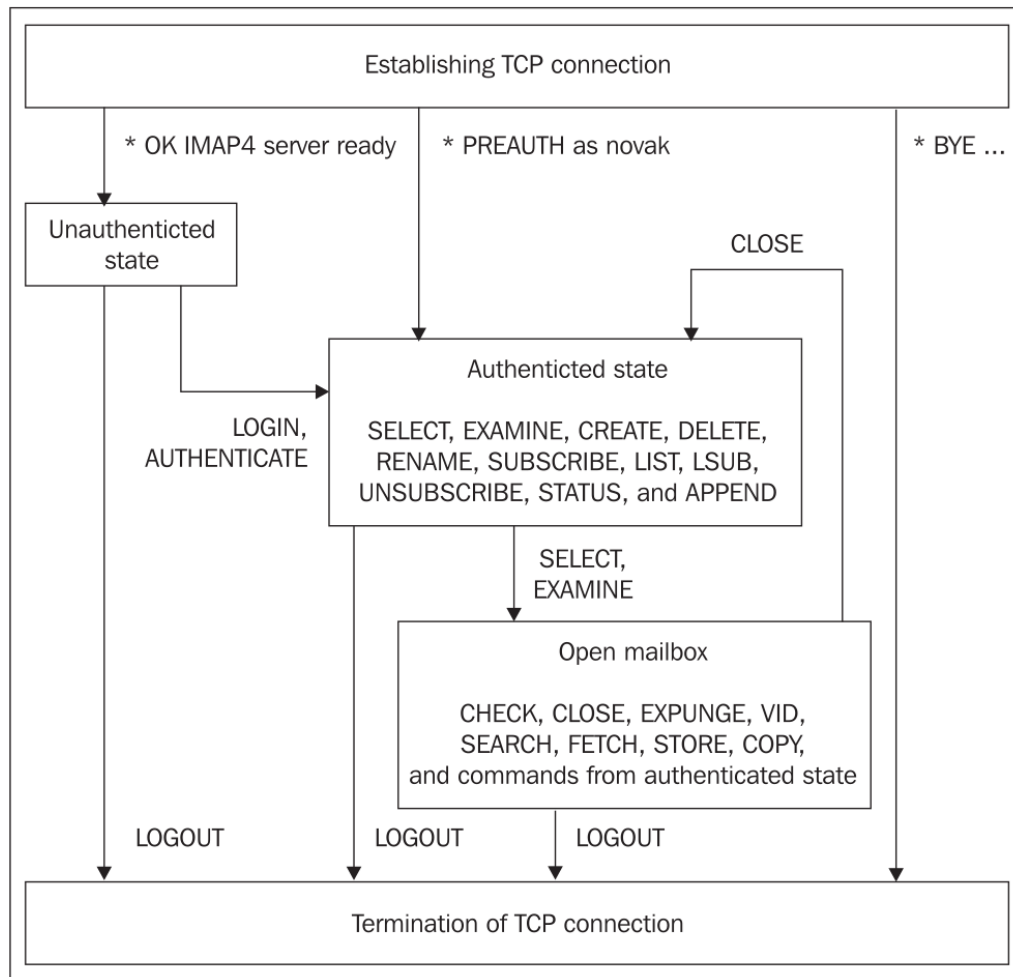


Figura 20: Estados del protocolo IMAP4

Como se pudo observar, POP3 tiene varias deficiencias ya que no permite al usuario organizar su correo en el servidor, no puede tener carpetas diferentes en el servidor y no permite verificar parcialmente el contenido del correo antes de descargarlo. Por su parte, IMAP4 ofrece las siguientes funciones adicionales:

- Un usuario puede verificar el encabezado del correo electrónico antes de descargarlo.
- Un usuario puede buscar una cadena específica de caracteres en el contenido del correo electrónico antes de descargarlo.
- Un usuario puede descargar parcialmente el correo electrónico. Esto es especialmente útil si el ancho de banda es limitado y el correo electrónico contiene contenido multimedia con altos requisitos de ancho de banda.
- Un usuario puede crear, eliminar o cambiar el nombre de los buzones en el servidor de correo.

- Un usuario puede crear una jerarquía de buzones en una carpeta para el almacenamiento de correo electrónico.

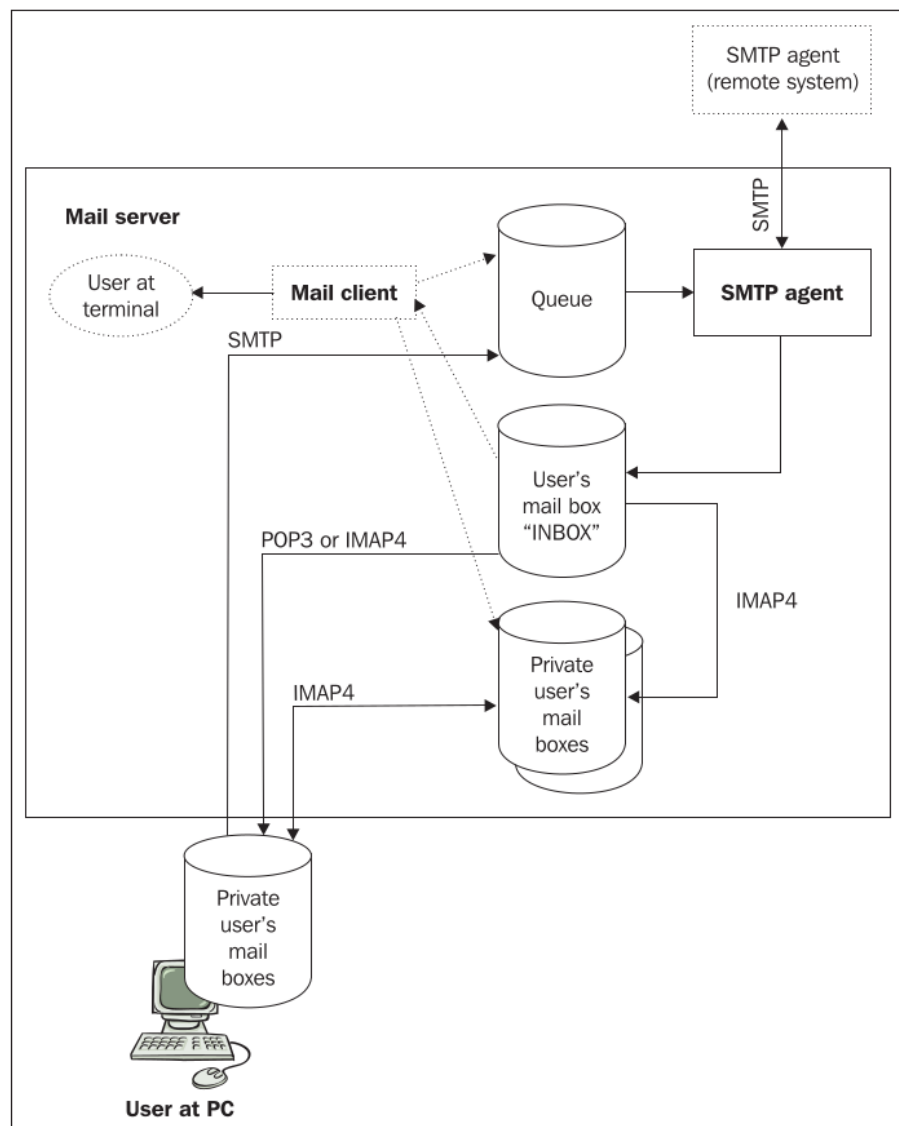


Figura 21: Protocolos POP3 e IMAP4

TAREA 1

Pregunta 1

¿En qué consiste el ESMTP y el S/MIME?

Pregunta 2

¿Cómo se puede agregar seguridad a los protocolos POP e IMAP?

Pregunta 3

¿Qué es un Webmail? ¿Cómo funciona?

Pregunta 4

¿Qué es SASL? ¿Cómo se puede utilizar en el correo electrónico?

Pregunta 5

Crea un script/programa en cualquier lenguaje de programación (Python, Java, Kotlin, Javascript, Typescript, Bash) que permita enviar un correo electrónico desde una cuenta Gmail/Outlook utilizando el protocolo SMTP a un destinatario específico.

Entregar en SiCCAAD archivo con código fuente y capturas de pantalla (video) del programa en ejecución.

Fecha de entrega: 20 de marzo de 2025

Referencias

- [1] C. Hunt, *TCP/IP Network Administration*. O'Reilly Media, 2002.
- [2] B. A. Forouzan, *Data Communications and Networking*. McGraw-Hill, 2003.
- [3] J. Bergstra y M. Burgess, *Handbook of Network and System Administration*. Elsevier, 2007.
- [4] D. E. Comer, *Internetworking with TCP/IP Vol. 1: Principles, Protocols, and Architecture*. Prentice Hall, 2000.
- [5] J. Edwards y R. Bramante, *Networking Self-Teaching Guide OSI, TCP/IP, LANs, MANs, WANs, Implementation, Management, and Maintenance*. Wiley, 2009.
- [6] D. Reynders y E. Wright, *Practical TCP/IP and Ethernet Networking*. Elsevier, 2003.
- [7] K. Siyan y T. Parker, *TCP/IP Unleashed*. Sams Publishing, 2002.
- [8] C. M. Kozierok, *The TCP/IP Guide*. Charles M. Kozierok, 2005.
- [9] L. Dostálek y A. Kabelová, *Understanding TCP/IP A clear and comprehensive guide to TCP/IP protocols*. Packt Publishing, 2006.
- [10] B. Hallberg, *Networking: A Beginner's Guide*. McGraw-Hill, 2001.