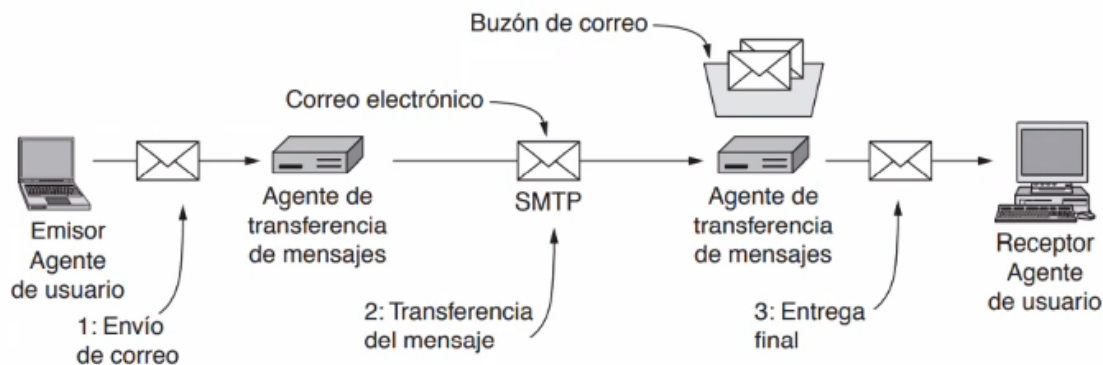


Email



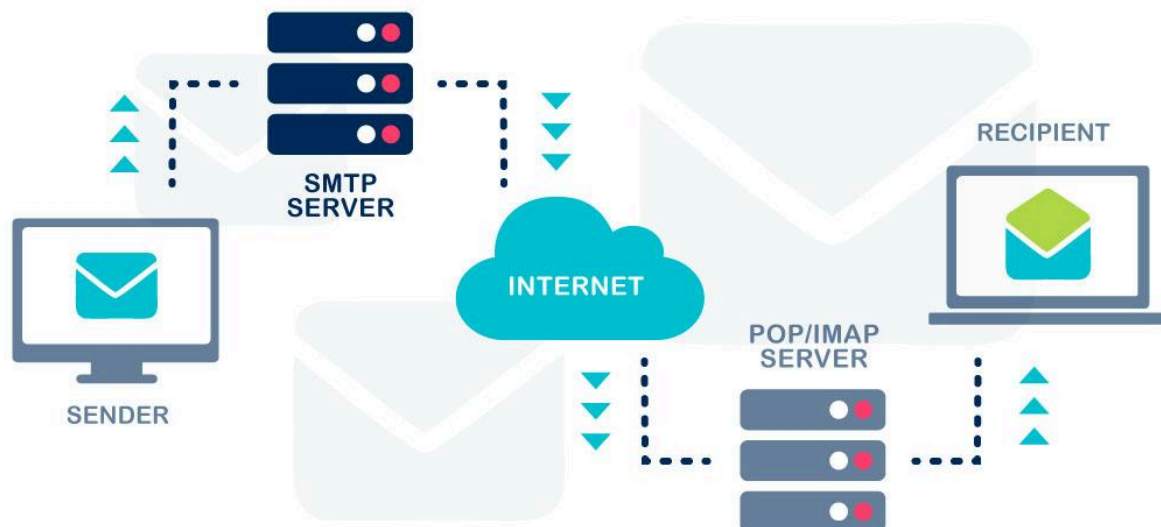
Se tiene un agente emisor de, actualmente el que conocemos son aquellos clientes que envían correos como outlook, webmail que es la interfaz gráfica de gmail. Se espera que haya un agente local que envía el correo a un mail service, que recibe el correo y se encarga de entregarlos, cuantos se habla de una interfaz web de envío de correo, normalmente el servidor del correo va a tener si base de datos, pero a nivel del correo electrónico, es decir al protocolo puro, ellos utilizan la carpeta, /var/spool de linux, que se meten los correos acá y tiene toda la codificación del correo electrónico, outlook, y gmail, no utilizan una carpeta si no que tienen una base de datos. Las interfaces en vez de utilizar el protocolo SMTP, utiliza un layer de web services que habla directamente con la base de datos y se lleva algún tipo de control. Mailx, sendmail, se pueden instalar en línea de comando para realizar envío de correos para aplicaciones como en shell, y poder enviar estos. En linux es postfix el servidor de correos por excelencia, creció en el opensource y se empezó a desarrollar y luego una empresa creó un fork y creció más tiene el OpenRelay en el 127.0.0.1.24, cuando se envía un correo al root, se pone el nombre del usuario al que se desea enviar y mete el correo en vars/pool, y muestra todos los correos disponibles, los servidores de CE almacenan los correos, antiguamente en un folder de la máquina, ahora en bases de datos. Los agentes locales siguen utilizando el protocolo de SMTP, puede que gmail aún utilice el protocolo original, es muy probable que lo que se utilice sea un webservice.

Un CE siempre lleva un to:, lleva un from:, con esta info el cliente de CE lo que hace es que se pega a un agente de transferencia de mensajes por medio del dominio del from, algo importante es que el correo electrónico tiene un send as:, que es que se puede cambiar el correo electrónico, siempre la resolución se va a hacer con el from, no con el send as, el agente de transferencia recibe el mensaje y lo mete en una especie de cola y es una espera que se da en tiempo real pero la realidad es que el protocolo no sea en tiempo real, pero con las velocidades que se tiene de internet, el tamaño del contenido, y es tan poco relevante que pasa muy rápido de un cliente a otro, el agente del correo electrónico inspecciona a quien se le debe responder y manda el correo hacia el servidor del usuario, cuando se hace esto, se debe imaginar que no es solo un servidor, si no algo más bien como un cluster. El correo después pasa por otros agentes de referencia y se puede imaginar qué es como una especie de forward. Es natural imaginar el correo como una colección de folders en un archivo, cuando se entrega al folder, el cliente que recibe el correo electrónico, revisa si hay en el servidor y

cuando hay alguno lo baja y lo pone dentro de la máquina. El correo electrónico buscaba simular el correo real, si la persona no se conectaba en cierto tiempo, el correo se borraba, estos protocolos son de la época, y el internet era de 64k por segundo, cerca de los noventas. El nivel de información que se podía manejar en la época era muy bajo. Se tenía en el email un encabezado, se tenía el cuerpo, el agente del usuario, la lista de spammers que es como una IA de etiquetado del correo, y se podía tener botnets. Las listas de spammers se pueden revisar online, en otros países es más común que se reporten los correos electrónicos como SPAM, es algo como cultura. El IP podría estar en la lista, pero podría ser que el ip se está siendo utilizado para enviar SPAM, recordar que el ip es compartido por más de una persona, por la sobre banda. De los botnets ya casi no se escucha mucho.

Mensaje formato: To, Cc, Bcc, From, Sender, Received, Return path, luego los encabezados: Date, Reply-To, Message-id, In-Reply-To, References, Keywords, Subject.

SMTPs



Imágen recuperada de: <https://serversmtp.com/what-is-smtp/>

SMTP corre en el puerto 25, es ASCII simple, acepta mensajes, en caso de no poder entregarlo, retorna notificación, para la conexión, cliente abre el puerto 25, el servidor indica que está listo para recibir mensajes y da su identidad, el cliente entonces envía HELO, MAILFROM, RCPT TO, DATA, QUIT.

MIME (Multipurpose Internet Mail Extension):

Tipo	Subtipos de ejemplo	Descripción
text	plain, html, xml, css	Texto en diversos formatos.
image	gif, jpeg, tiff	Imágenes.
audio	basic, mpeg, mp4	Sonidos.
video	mpeg, mp4, quicktime	Películas.
model	vrml	Modelo 3D.
application	octet-stream, pdf, javascript, zip	Datos producidos por aplicaciones.
message	http, rfc822	Mensaje encapsulado.
multipart	mixed, alternative, parallel, digest	Combinación de múltiples tipos.

Internet Message Access Protocol (IMAP):

UNSUBSCRIBE	Elimina carpeta del conjunto activo.
LIST	Lista las carpetas disponibles.
LSUB	Lista las carpetas activas.
STATUS	Obtiene el estado de una carpeta.
APPEND	Agrega un mensaje a una carpeta.
CHECK	Obtiene un punto de verificación de una carpeta.
FETCH	Obtiene mensajes de una carpeta.
SEARCH	Busca mensajes en una carpeta.
STORE	Altera las banderas de los mensajes.
COPY	Crea una copia de un mensaje en una carpeta.
EXPUNGE	Elimina los mensajes marcados para eliminación.
UID	Emite comandos mediante el uso de identificadores únicos.
NOOP	No hace nada.
CLOSE	Elimina los mensajes marcados con banderas y cierra la carpeta.
LOGOUT	Cierra la sesión y la conexión.

World Wide Web (WWW)

1989 CERN, W3C, Desarrollo de web, estandarizar protocolos e interoperabilidad.

Webpages, Links, Vannevar Bush creo lo de los links que definio el internet. Browser.

HTTP y HTML no es lo mejor que se puede tener, al ser un mensaje textual, se gasta mucho ancho de banda para realizar envío de datos, fue tan exitoso que todo mundo lo utilizo y es muy interoperable y es texto.

HTTP mueve información, páginas dinámicas y estáticas, scripts web, Dynamic HTML.

JavaScript, JVM Applets. MIME, Cookies, HTML, CSS, Common Gateway Interface,

XML -> SSO, JSON. SOAP, Caches, CRUD -> API

Método	Descripción
GET	Leer una página web.
HEAD	Leer el encabezado de una página web.
POST ➕	Adjuntar a una página web.
PUT	Almacenar una página web.
DELETE	Eliminar la página web.
TRACE	Repetir la solicitud entrante
CONNECT	Conectarse a través de un proxy
OPTIONS	Consultar las opciones para una página

Código	Significado	Ejemplos
1xx	Información	100 = el servidor acepta manejar la solicitud del cliente.
2xx	Éxito	200 = la solicitud es exitosa; 204 = no hay contenido.
3xx	Redirección	301 = se movió la página; 304 = la página en caché aún es válida.
4xx	Error del cliente	403 = página prohibida; 404 = no se encontró la página.
5xx	Error del servidor	➕ 500 = error interno del servidor; 503 = intentar más tarde.

URLs

Nombre	Se usa para	Ejemplo
http	Hipertexto (HTML).	http://www.ee.uwa.edu/~rob/
https	Hipertexto con seguridad.	https://www.bank.com/accounts/
ftp	FTP.	ftp://ftp.cs.vu.nl/pub/minix/README
file	Archivo local.	file:///usr/suzanne/prog.c
➕mailto	Enviar correo electrónico.	mailto:JohnUsuario@acm.org
rtsp	Medios de flujo continuo.	rtsp://youtube.com/montypython.mpg
sip	Llamadas multimedia.	sip:eva@adversario.com
about	Información del navegador.	about:plugins

Algunos protocolos que se utilizan

Imágenes obtenidas de la presentación de la clase del 6/6/25