Práctica 3 - Protocolos en la capa de aplicación TCP-IP

Instrucciones

1. Elegir un protocolo de la capa de aplicación orientado a conexión y un protocolo de la capa de aplicación no orientado a conexión.
2. Reutilizar las aplicaciones cliente y servidor creadas en la práctica 3 para gestionar múltiples conexiones. Usar la implementación del protocolo de la capa de transporte para mandar los mensajes de la capa de aplicación.

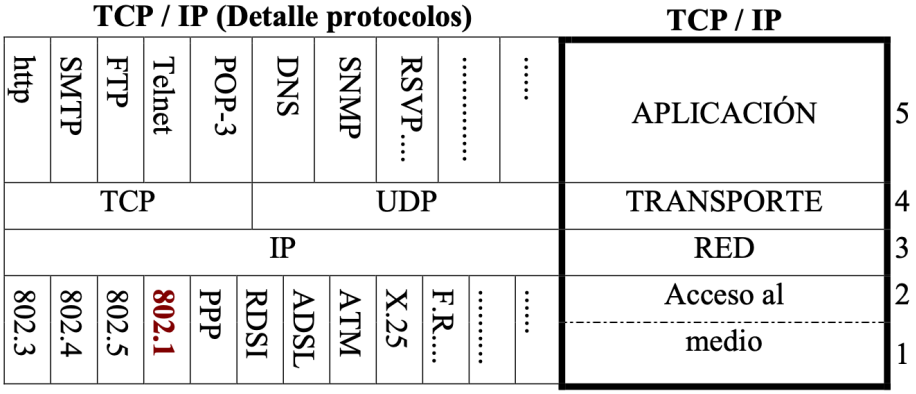
3. Implementar desde cero, (sin usar ningún módulo o librería) las aplicaciones cliente y servidor que simulen el intercambio de mensajes de una transacción de los protocolos de la capa de aplicación. Se debe apegar a los estándares mencionados en este documento.

Introducción

La jerarquía de los protocolos TCP/IP está organizada como una pila de capas o niveles, cada una construida a partir de la que está debajo de ella. Las capas pueden ofrecer dos tipos de servicio a las capas que están sobre ellas: orientado a conexión y no orientado a conexión.

La capa de aplicación está construida sobre los protocolos de la capa de transporte, así que podemos encontrar protocolos de aplicación que usen TCP o protocolos de aplicación que usan UDP.

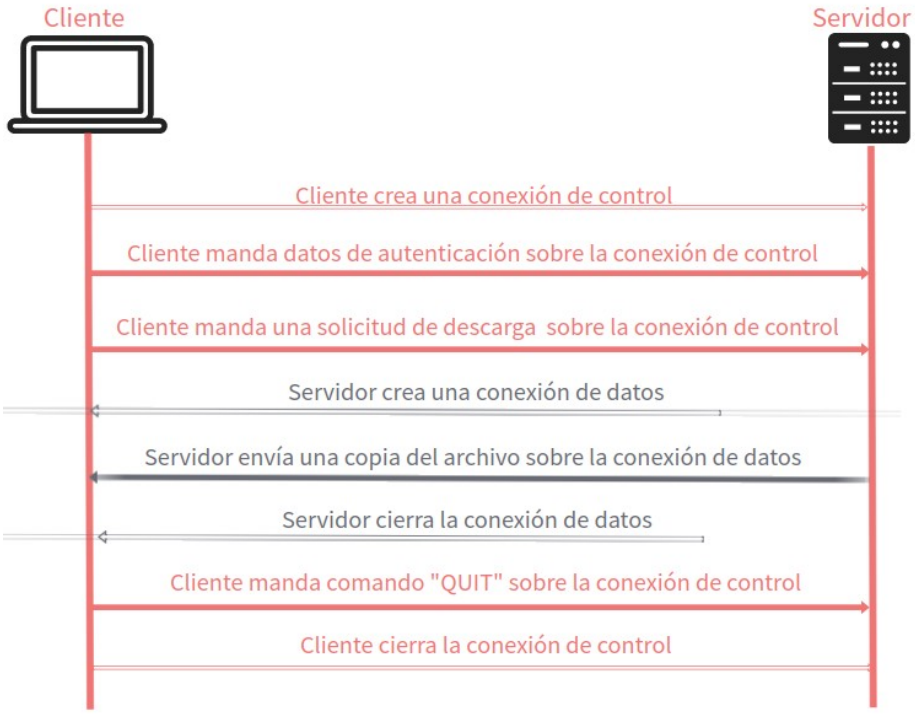
La capa de aplicación contiene varios protocolos que los usuarios requieren con frecuencia. Un protocolo de aplicación de amplio uso es el HTTP (Protocolo de transferencia de Hipertexto), que es la base del World Wide Web. Cuando un navegador desea una página Web, utiliza este protocolo para enviar al servidor el nombre de dicha página. A continuación el servidor devuelve la página. Otros protocolos de aplicación se utilizan para la transferencia de archivos, correo electrónico y noticias en la red.



FTP

El Protocolo de transferencia de archivos (File Transfer Protocol o FTP) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo. Sigue el estándar RFC 959 y tiene las extensiones para IPv6 y NATs: RFC 2428.

Uno de los aspectos más importantes de FTP radica en la interacción entre el cliente y el servidor. El cliente establece una conexión a un servidor FTP enviando una serie de peticiones que el servidor responde. A diferencia de HTTP, FTP no responde en la misma conexión que el cliente manda las peticiones. En su lugar, la conexión que el cliente crea, llamada conexión de control, es reservada para comandos. Cada vez que el servidor necesita descargar o cargar un archivo, el servidor abre una nueva conexión. Dicha conexión es llamada conexión de datos.



Algoritmo

Dado:
Una conexión de control FTP

Se debe lograr:
Transmisión de datos en una conexión FTP

Método:
Cliente envía dato de autenticación USER usando una conexión de control
Servidor recibe la petición y contesta código según estado
Cliente envía dato de autenticación PASS usando una conexión de control
Servidor recibe la petición y contesta código según estado
if (Autenticación de usuario es correcta) {
    Cliente envía una petición por un archivo en específico usando una conexión de control
    Servidor recibe la petición
    Cliente asigna un puerto local al protocolo, llamemoslo X
    Cliente enlaza el puerto X y se prepara para aceptar una conexión
    Cliente envía “PUERTO X” al servidor por la conexión de control
    Servidor recibe el comando PUERTO y la solicitud de un elemento
    Cliente espera por la conexión de datos en el puerto X y la acepta
    Servidor crea una conexión de datos al puerto X al cliente
    Servidor envía el archivo solicitado usando la conexión de datos
    Servidor cierra la conexión de datos.
}
Cliente envía “QUIT” al servidor para finalizar la conexión de control
Cliente cierra la conexión de control

Nota: No se implementa el manejo de directorios (sistema de archivos)

Escenario típico de FTP

Un usuario en el host Client quiere transferir archivos de/hacia el host Server
Client connect server 127.0.0.1 port 21
Server 220 Service ready <CRLF>
Client USER <SP> Doe<CRLF>
Server 331 User name ok, need password<CRLF>.
`Client PASS <SP> mumble<CRLF>`
Server 230 User logged in<CRLF>
(cliente carga el archivo holi.jpg en el buffer)
`Client STOR <SP>holi.jpg<CRLF>`
Server 150 File status okay; about to open data connection <CRLF>
Client PORT <SP> 20 <CRLF>
Server 125 Data connection already open; transfer starting. <CRLF>

Server 226 Closing data connection, file transfer successful<CRLF>.
Client QUIT <CRLF>
Server closes all connections.

`Nota: Se deben implementar sólo los comandos FTP para lograr una transferencia (envío y recepción) de archivos entre las aplicaciones cliente-servidor.`

SMTP

El **protocolo para transferencia simple de correo** (**Simple Mail Transfer Protocol** o **SMTP**) es un protocolo de red utilizado para el intercambio de mensajes de correo electrónico. Fue definido inicialmente en agosto de 1982 por el RFC 821 (para la transferencia) y el RFC 822 (para el mensaje). Son estándares oficiales de Internet que fueron reemplazados respectivamente por el RFC 2821 y el RFC 2822, que a su vez lo fueron por el RFC 5321 y el RFC 5322.

Algoritmo

Dado
Una transacción de correo desde un usuario a otro

Se debe lograr:
La transmisión del mensaje al destinatario

Método
El usuario recurre a la interfaz de la aplicación de correo y genera un mensaje de correo para el usuario x@destino.com

La interfaz de la aplicación de correo del usuario encola el mensaje para ser transferido.

El programa de transferencia de correo del usuario examina la cola de correos y encuentra el mensaje.

El programa de transferencia de correo abre una conexión a destino.com
El programa de transferencia de correo usa SMTP para transferir el mensaje
El programa de transferencia de correo cierra la conexión
El servidor de correos en destino.com recibe el mensaje y pone una copia en la carpeta “bandeja de entrada” del usuario X
El usuario X de destino.com ejecuta la interfaz de la aplicación de correo, que despliega la “bandeja de entrada” del usuario, incluyendo el nuevo mensaje.

Como el algoritmo indica, el software de correo está dividido en dos piezas separadas conceptualmente:
La interfaz de la aplicación de correo
El programa de transferencia de correo.

El usuario invoca la interfaz de la aplicación de correo de manera directa. Esta interfaz provee mecanismos que permiten al usuario componer y editar los mensajes salientes así como leer y procesar los mensajes entrantes. La interfaz de aplicación de correo no actúa como cliente o como servidor, y no transfiere mensajes a otros usuarios. En su lugar, la interfaz de aplicación lee los mensajes la bandeja de entrada del usuario y deposita los mensajes salientes en la cola de mensajes de salida. De manera separada, el programa conocido como “programa de transferencia de correo” y el servidor de correos atienden las transferencias de correos. El programa de transferencia de correo actúa como un cliente

para enviar un mensaje al servidor de correo en la computadora destino. El servidor de correo acepta mensajes entrantes y las deposita en la bandeja de entrada del usuario correspondiente.

Transacción de correos:

Server: 220 destino.com Simple Mail Transfer Service Ready
Client: HELO destino.com
Server: 250 OK
Client: MAIL FROM:<John_Q_Smith@destino.com>
Server: 250 OK
Client: RCPT TO:<Mathew_Doe@somewhere.com>
Server: 550 No such user here
Client:RCPT TO:<Paul_Jones@destino.com>
Server: 250 OK
Client:DATA
Server:354 Start mail input; end with <CR><LF>.<CR><LF>
Client: ...sends body of mail message, which can contain
Client: ...arbitrarily many lines of text
Client: <CR><LF>.<CR><LF>
Server: 250 OK
Client:QUIT
Server: 221 destino.com closing transmission channel

Nota: Sólo se enviará texto.
Nota 2: Se debe crear una carpeta por cada usuario, esta carpeta será la “bandeja de entrada”

DNS

El sistema de nombres de dominio (Domain Name System o DNS) es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP como Internet o una red privada. Este sistema asocia información variada con nombres de dominio asignados a cada uno de los participantes. Su función más importante es "traducir" nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente. Ver rcf 1033, rcf 1035, rfc 1034 rfc 1591, ProtocoloDNS

El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

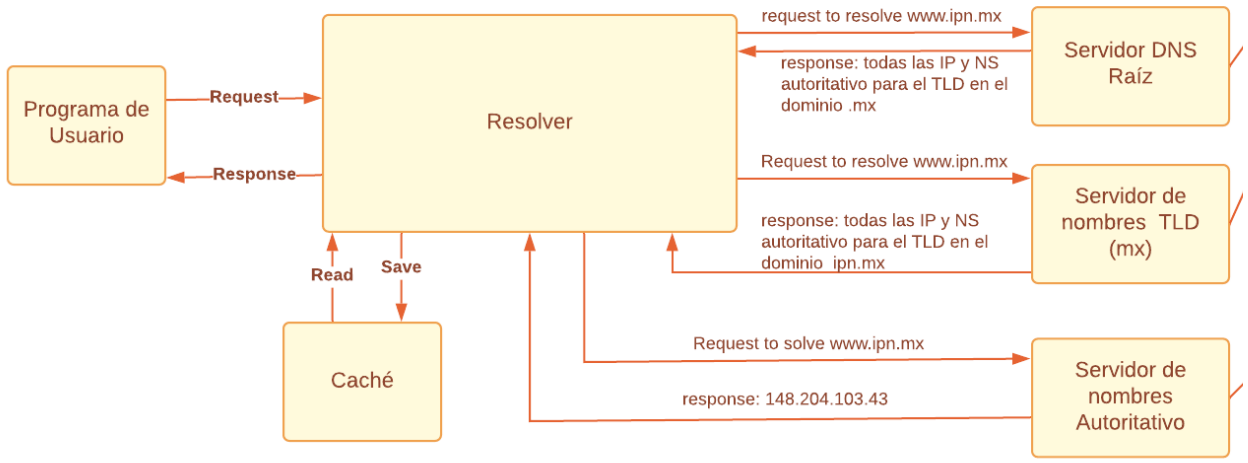
Visita el sitio https://www.verisign.com/en_US/website-presence/online/how-dns-works/index.xhtml

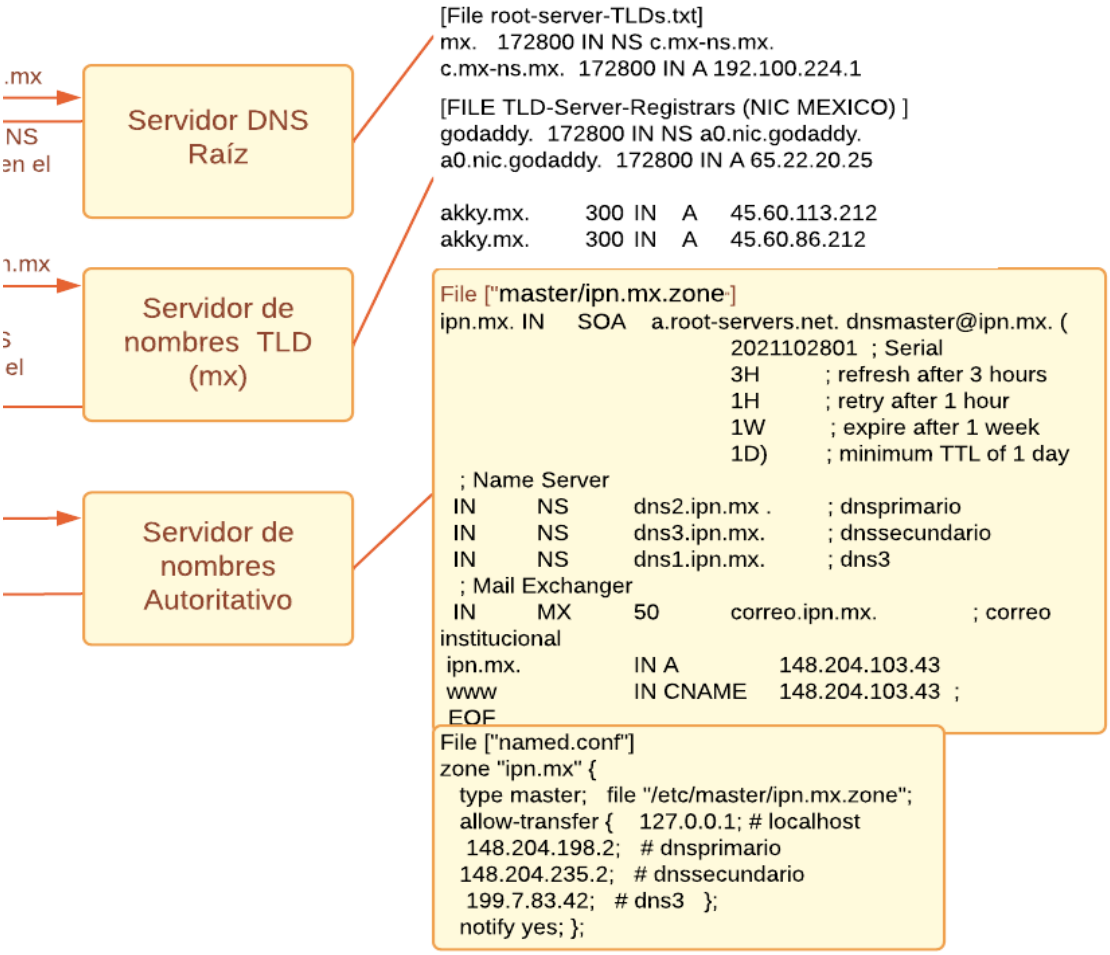
Ver video https://www.youtube.com/watch?v=LqSqrtxrW7w

Algoritmo

Dado:
Un mensaje de solicitud de un solucionador de nombres DNS
Se debe lograr:
Un mensaje de respuesta que contenga la dirección IP.
Método:

1. El usuario teclea la URL *http://www.ipn.mx*
2. El programa de usuario envía la pregunta al resolver: ¿Cuál es la IP de *www.ipn.mx*?
3. El resolver del DNS mira si la respuesta está en su caché, pero no la encuentra.
4. El resolver del DNS le envía la pregunta a uno de los servidores raíz. Ver rootServer
5. El servidor raíz responde con la lista de IP (referencias) de los servidores de nombres autoritarios del siguiente nivel al que él es autoritario. El es autoritario en el nivel raíz (el punto final de un FQDN), la lista es del nivel ccTLD *.mx*. Ver TLDs
6. El resolver del DNS elige una IP de la lista recibida, y le envía la pregunta a dicho servidor de nombres que posee el fichero de zona del ccTLD *.mx*.
7. El servidor DNS del ccTLD *.mx* responde por tanto con la lista de IP (referencias) de los servidores DNS autoritarios del siguiente nivel, el SLD *.ipn.mx*. Ver Registars
8. El resolver del DNS elige una IP de la lista recibida, y le envía la pregunta a dicho servidor de nombres, que ya es un servidor DNS autoritario del dominio *.ipn.mx*, por lo que dispondrá del fichero de zona del dominio consultado.
9. El servidor DNS autoritativo del dominio *.ipn.mx.* consulta el fichero de zona y devuelve la respuesta al nombre *www.ipn.mx*, que estará formada por los RR A y los RR CNAME si hubiera alguno relacionado. Ver ZoneFile y Whois
10. El resolver del DNS envía la respuesta completa (RR A y RR CNAME) al programa del usuario.
11. El programa del usuario muestra el primer registro A, es decir, la primera IP.


Programa de Usuario
Request
Response
Resolver
Read
Save
Caché
request to resolve www.ipn.mx
response: todas las IP y NS autoritativo para el TLD en el dominio .mx
Servidor DNS Raíz
Request to resolve www.ipn.mx
response: todas las IP y NS autoritativo para el TLD en el dominio ipn.mx
Servidor de nombres TLD (mx)
Request to solve www.ipn.mx
response: 148.204.103.43
Servidor de nombres Autoritativo


Servidor DNS Raíz
Servidor de nombres TLD (mx)
Servidor de nombres Autoritativo
[File root-server-TLDs.txt]
mx. 172800 IN NS c.mx-ns.mx.
c.mx-ns.mx. 172800 IN A 192.100.224.1
[FILE TLD-Server-Registrars (NIC MEXICO)]
godaddy. 172800 IN NS a0.nic.godaddy.
a0.nic.godaddy. 172800 IN A 65.22.20.25
akky.mx. 300 IN A 45.60.113.212
akky.mx. 300 IN A 45.60.86.212
File ["master/ipn.mx.zone"]
ipn.mx. IN SOA a.root-servers.net. dnsmaster@ipn.mx. (
2021102801 ; Serial
3H ; refresh after 3 hours
1H ; retry after 1 hour
1W ; expire after 1 week
1D) ; minimum TTL of 1 day
; Name Server
IN NS dns2.ipn.mx . ; dnsprimario
IN NS dns3.ipn.mx. ; dnssecundario
IN NS dns1.ipn.mx. ; dns3
; Mail Exchanger
IN MX 50 correo.ipn.mx. ; correo institucional
ipn.mx. IN A 148.204.103.43
www IN CNAME 148.204.103.43 ;
EOF
File ["named.conf"]
zone "ipn.mx" {
type master; file "/etc/master/ipn.mx.zone";
allow-transfer { 127.0.0.1; # localhost
148.204.198.2; # dnsprimario
148.204.235.2; # dnssecundario
199.7.83.42; # dns3 };
notify yes; };

SNMP

El protocolo simple de administración de red (SNMP), publicado en 1988, fue diseñado para proporcionar una implementación sencilla para la gestión de redes de múltiples fabricantes de enrutadores, servidores, estaciones de trabajo y otros recursos de la red.
La especificación de SNMP: Define un protocolo para el intercambio de información entre uno o más gestores y varios de agentes. Proporciona una estructura para dar formato y almacenamiento de información de gestión. Define una serie de variables de información de gestión de propósito general, u objetos.

Conceptos básicos de SNMP

En esta sección se describen los elementos principales del protocolo. Primero se hablará de la arquitectura de administración de red, posteriormente se estudiarán los métodos que usa el protocolo SNMP para intercambiar información entre el gestor y el agente.

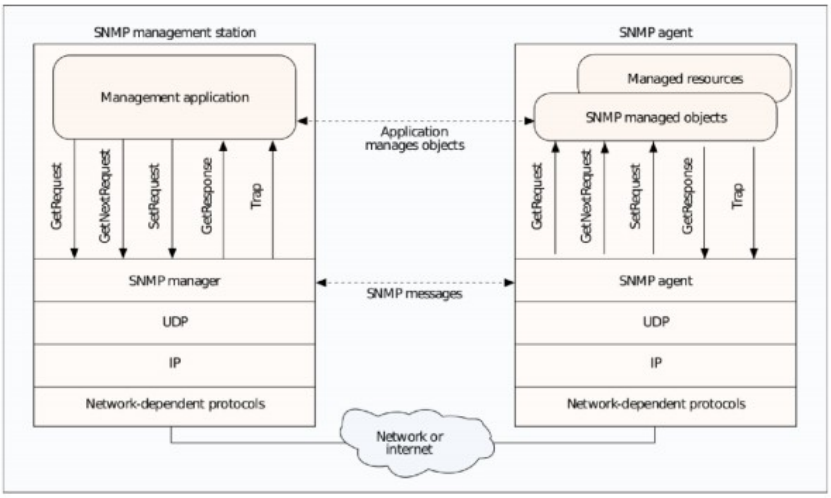
Arquitectura de administración de red

El modelo de administración de red que es usado por SNMP incluye los siguientes elementos:

- Gestor
- Agente
- Base de Información de Administración (Management information base MIB)
- Protocolo de administración de red

El gestor y el agente están ligados por el protocolo de administración de red que incluye los siguientes mensajes:

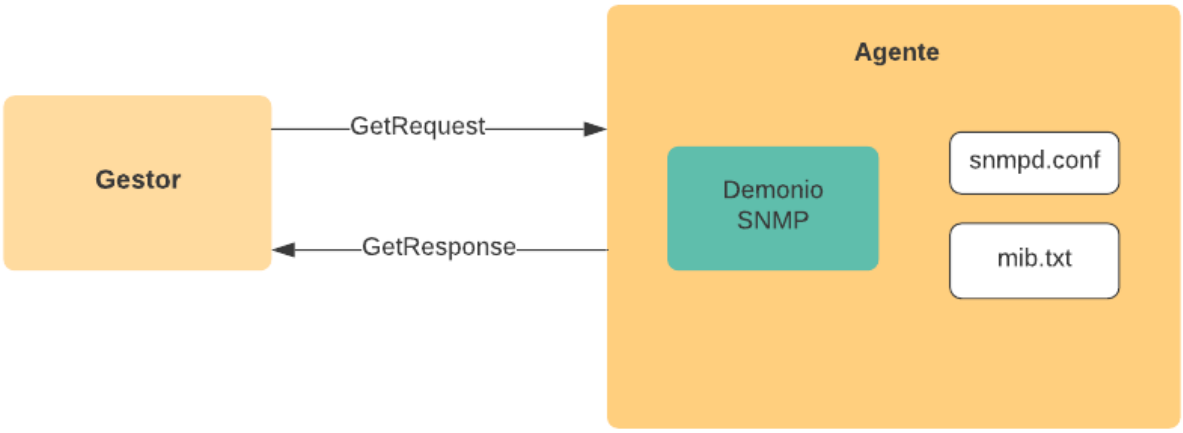
- GET : Permite al gestor obtener el valor de un objeto del agente
- SET: Permite al gestor modificar el valor de un objeto del agente
- TRAP: Permite que el agente notifique al gestor eventos significativos.



Algoritmo

Dado:
Un mensaje de solicitud de información SNMP
Se debe conseguir:
La información de administración del recurso solicitado
Método

El agente inicia la recepción de solicitudes iniciando al demonio SNMPD.
El demonio SNMPD carga los datos de configuración del archivo snmpd.conf para obtener datos de control de acceso (comunidad), sólo responderá las solicitudes de información que tengan los permisos de acceso correctos (comunidad correcta y permiso de lectura/escritura).
El demonio SNMPD escucha el puerto 161 por solicitudes de información.
El gestor construye el OID del objeto que quiere consultar.
El gestor manda una solicitud GETREQUEST al agente que incluye el OID y la COMUNIDAD
El demonio SNMPD recibe la solicitud GETREQUEST y valida la información de control de acceso (COMUNIDAD y permiso de lectura/escritura)
si (la comunidad es correcta){
    El demonio SNMPD consulta el archivo MIB.txt para conocer/modificar la información del OID
}
El demonio SNMPD responde la solicitud al agente con el código correspondiente y el mensaje del OID (var binds)
El gestor recibe la respuesta y la muestra al usuarios



Nota: Sólo se implementan las operaciones GET y SET de SNMP.