MODELO OSI - CAPAS 4 A 7 NAT Y PPROXY

1 OBJETIVO

El objetivo de la presente clase es presentar el funcionamiento de las capas superiores del modelo OSI 4 a 7, ver cuales son sus características de funcionamiento de cada una de ellas, las soluciones que implementan y como interactúan con los servicios cliente, por ejemplo al tratar de vincular redes privadas con las públicas mediante NAT o Proxy.

2 INTRODUCCIÓN

Hasta este momento hemos visto que la capa 3 del modelo OSI nos provee de un mecanismo para poder identificar distintas redes y host dentro de estas de forma inequívoca, esto también nos permite realizar comunicaciones punto a punto sin conexión utilizando como enlace a los routers. Si avanzamos un poco sobre lo visto, podríamos analizar otros escenarios tales como, que es lo que sucede cuando un router recibe una solicitud para una IP pública desde una red privada, que pasa con un paquete cuando su información es alterada durante el transporte, o que criterio utilizamos para el envío de grandes volúmenes de información.

Es evidente que todo esto no es posible controlarlo desde la capa de 3 y surge la necesidad de un conjunto de soluciones específicas para cada problema, estas respuestas se encuentran implementadas en las capas superiores del modelo OSI, las que inmediatamente comenzaremos a desarrollar.

3 CAPA DE TRANSPORTE (CAPA 4):

Es la encargada de llevar a cabo una comunicación punto a punto entre máquinas brindando una comunicación confiable, pude encargarse o no de la recuperación de datos en caso de corrupción de los mismos, dependiendo del protocolo utilizado y el controlar el flujo sobre la conexión establecida. Esto asegurará que todos los paquetes llegarán intactos y en un correcto orden a destino. Además esta capa efectúa la segmentación de la información en unidades más pequeñas cuando el host debe a transmitir, y reensamblar los segmentos recibidos desde otro host.

Si deseamos insertar al protocolo TCP/IP dentro del modelo OSI, podemos ver que dentro de la capa 4 se pueden utilizar dos protocolos TCP o UDP, ambos incluidos dentro de TCP/IP. La utilización de uno u otro depende del tipo de aplicación que utilicemos para enviar la información o como se dice en otras oportunidades, orientado con conexión o sin conexión, métodos que describiremos mas adelante.

3.1 TCP

La función protocolo TCP (Transmission Control Protocol – Protocolo de Control de Transmisión) consiste en ofrecer un servicio de envío y recepción de datos orientado a conexión, esto significa que un vez encontrado el destino mediante IP, mediante el protocolo TCP se implementa un meca-

nismo de entrega de paquetes que permite el seguimiento, arribo y la integridad de estos en destino. Esto significa tener una conexión confiable con un flujo continuo, seguro, sin perdida de datos y similar a una conexión punto a punto dedicado.

Por lo tanto podemos decir que este protocolo provee confiabilidad, garantizando la correcta entrega de datos. Además el protocolo TCP está diseñado para controlar el envío y recepción de segmentos TCP a fin de evitar momentos de congestión en la red efectuando control de flujo.

En la próxima figura podemos ver el formato de un segmento TCP.

BITS **0** **16** **32**			
Puerto TCP origen			Puerto TCP destino
Número de secuencia			
Número de acuse de recibo			
HLEN	Reservado	Bits código	Ventana
Suma de verificación			Puntero de urgencia
Opciones (si las hay)			
Datos … …			

- **Puerto origen** (16 bits). Puerto de la máquina origen. Al igual que el puerto destino es necesario para identificar la conexión actual.
- **Puerto destino** (16 bits). Puerto de la máquina destino.
- **Número de secuencia** (32 bits). Indica el número de secuencia del primer byte que trasporta el segmento.
- **Número de acuse de recibo** (32 bits). Indica el número de secuencia del siguiente byte que se espera recibir. Con este campo se indica al otro extremo de la conexión que los bytes anteriores se han recibido correctamente.
- **HLEN** (4 bits). Longitud de la cabecera medida en múltiplos de 32 bits (4 bytes). El valor mínimo de este campo es 5, que corresponde a un segmento sin datos (20 bytes).
- **Reservado** (6 bits). Bits reservados para un posible uso futuro.

- **Bits de código** o indicadores (6 bits). Los bits de código determinan el propósito y contenido del segmento. A continuación se explica el significado de cada uno de estos bits (mostrados de izquierda a derecha) si está a 1.
  - **URG.** El campo *Puntero de urgencia* valida la información del puntero de urgencia.
  - **ACK.** Este campo indica si esta siendo usado el campo acuse de recibo.
  - **PSH.** Indica si se necesita velocidad de transmisión alta.
  - **RST.** Indica reseteo de la conexión actual.
  - **SYN.** Sincronización de los números de secuencia. Se utiliza al crear una conexión para indicar al otro extremo cual va a ser el primer número de secuencia con el que va a comenzar a transmitir.
  - **FIN.** Indica al otro extremo que la aplicación ya no tiene más datos para enviar. Se utiliza para solicitar el cierre de la conexión actual.
- **Ventana** (16 bits). Indica el número de bytes que el emisor del segmento está dispuesto a aceptar por parte del destino.
- **Suma de verificación** (24 bits). Suma de comprobación de errores del segmento actual.
- **Puntero de urgencia** (8 bits). Se utiliza cuando se están enviando datos urgentes que tienen preferencia sobre todos los demás e indica el siguiente byte del campo *Datos* que sigue a los datos urgentes. Esto le permite al destino identificar donde terminan los datos urgentes. Nótese que un mismo segmento puede contener tanto datos urgentes (al principio) como normales (después de los urgentes).
- **Opciones** (variable). Si está presente define el tamaño máximo de segmento que será aceptado.
- **Relleno.** Se utiliza para que la longitud de la cabecera sea múltiplo de 32 bits.
- **Datos.** Información que envía la aplicación.

Cuando se utiliza TCP el primer paso que realiza es enviar un paquete con el número de secuencia y los bits de SYN para comenzar transmisión, por cada envío realizado se solicitará una confirmación de la recepción de los datos en el destino, si se detecta un error en la recepción de los datos, se solicitaría el reenvío de la información, asegurando de esta forma la integridad de la misma.

3.2 UDP

El protocolo UDP (User Datagram Protocol – Protocolo de Datagrama de Usuario) no utiliza mecanismos para asegurar la conexión. Como podemos apreciar no tiene ningún mecanismo para realizar verificaciones, solo tiene los puertos de origen y destino para realizar la entregas y un suma de verificación como mecanismo de integridad. En este caso son las aplicaciones y no el protocolo UDP las que se responsabilizan de la verificación de los datos en la transmisión y recepción.

Por estos motivos es que se lo llama servicio sin conexión y de baja fiabilidad, no garantiza las entregas, duplicaciones y llegadas en orden en destino.

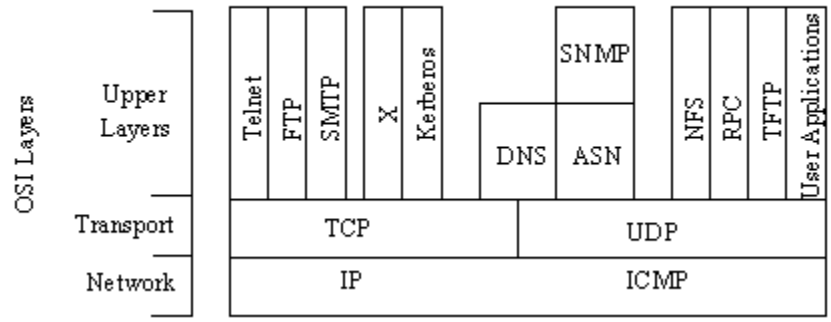
En la próxima figura vemos el formato de un segmento UDP:

BITS 0 16 32	
Puerto UDP origen	Puerto UDP destino
Longitud mensaje UDP	Suma verificación UDP
Datos	

Los dos primeros sectores tienen el mismo uso, y la diferencia es que en este protocolo solo se declara el tamaño del paquete y el único mecanismo de seguridad es una suma de comprobación o *Checksum.*

La no utilización de control de recepción, parece raro y hay más de una explicación para esto, una de ellas es evitar la sobrecarga de la red con inicios, confirmaciones y reenvíos. Otra podría ser para agilizar el transporte de datos como video, para ejemplificar esto podemos plantear lo que sucedería durante una videoconferencia, si un frame o parte del mismo se perdiera, la comunicación se detendría y se sería obligatoria la retransmisión de los datos perdidos con la consiguiente demora en la próxima recepción, afectando la fluidez de las imágenes y una gran demora en la comunicación.

En la próxima figura podemos ver cuales son las aplicaciones que utilizan los protocolos TCP y UDP de forma predeterminada.



Telnet - RemoteLogin
FTP - File Transfer Protocol
SMTP - Simple Mail Transfer Protocol
X - X Windows System
Kerberos - Security
DNS - Domain Name System
ASN - Abstract Syntax Notation
SNMP - Simple Network Management Protocol

NFS - Network File Server
RPC - Remote Procedure Calls
TFTP - Trivial File Transfer Protocol
TCP - Transmission Control Protocol
User Datagram Protocol
IP - Internet Protocol
ICMP - Internet Control Message Protocol

3.3 PUERTOS

Así como la capa de red utiliza a la dirección IP para realizar las conexiones, la capa de transporte implementa al puerto como un sistema de identificación para llegar a un destino específico entre varios dentro de un mismo host. Los puertos utilizan números para su identificación, este consta de 16 bits que da como resultante 65535 puertos en total los disponibles para entablar comunicaciones.

La designación de números no es aleatoria y se encuentra regulada por la organización IANA (Internet Assigned Numbers Authority – Autoridad de Números Asignados en Internet), esta divide a los puertos en dos grandes grupos como sigue a continuación:

- 0 a 255: se utilizan para aplicaciones servidor. También llamados well - known – bien conocidos.
  - 256 a 1023: son utilizados por organizaciones que identifican productos de Internet.
- 1024 o superior: se asignan de forma dinámica a las aplicaciones cliente que utilicen una aplicación de red. Estos ingresan en la categoría de los llamados puertos registrados (denominación IANA).

Los números asignados a las aplicaciones más comunes del tipo público son:

TCP			UDP	
HTTP	FTP 20/21	POP 110	DNS 53	TFTP 69
SMTP 25	DNS 53		SNMP 161/162	DNS 53

Esta numeración corresponde al proveedor del servicio y no a la máquina que lo solicita (máquina de usuario), estas utilizan el rango 1024 o superior.

En la figura anterior se pueden observar que mas de un servicio utiliza dos números, por ejemplo el FTP que utiliza el 21 para solicitar una petición y el 20 para realizar la descarga, con el servicio DNS sucede que utiliza el número 53 sobre TCP cuando se realiza una comunicación entre servidores y también utiliza el número 53 pero sobre UDP cuando se realiza la comunicación desde un cliente a un servidor.

Las próximas capas serán descriptas en forma inversa para comprender mejor su funcionamiento, comenzaremos por la Aplicación y así sucesivamente.

4 APLICACIÓN (CAPA 7)

La capa de aplicación es la capa del modelo OSI más cercana al usuario, y está relacionada con las funciones de más alto nivel, proporcionando soporte a las aplicaciones o actividades del sistema, suministrando servicios de red a las aplicaciones del usuario y definiendo los protocolos usados por las aplicaciones individuales. Esta capa es la que contiene un soporte de comunicaciones para poder enviar o recibir un tipo de información específica, cuando nos referimos a una aplicación lo hacemos a las de red, pero hoy en día las aplicaciones de las PC han evolucionado mucho y también integran herramientas para poder enviar la información generada vía mail solo por citar un ejemplo.

Difiere de las demás capas debido a que no proporciona servicios a ninguna otra capa OSI, sino solamente a aplicaciones que se encuentran fuera del modelo (procesadores de texto, hojas de cálculo, navegadores web, etc.).

La capa de aplicación establece la disponibilidad de los diversos elementos que deben participar en la comunicación, sincroniza las aplicaciones que cooperan entre sí y establece acuerdos sobre los procedimientos de recuperación de errores y control de la integridad de los datos.

Las aplicaciones de red más comunes son las de correo electrónico, navegación web y transferencia de archivos, mientras que las aplicaciones del SO como el procesador de texto hoy en día incluyen funciones de enviar a través de correo electrónico, presentaciones gráficas, planilla de cálculos, Internet Explorer u otro navegador.

Los procesos de las aplicaciones se comunican entre sí por medio de entidades de aplicación propias, estando éstas controladas por protocolos específicos de la capa de aplicación, que a su vez utilizan los servicios de la capa de presentación, situada inmediatamente debajo en el modelo.

5 PRESENTACIÓN (CAPA 6)

La capa de presentación proporciona sus servicios a la capa de aplicación, garantizando que la información que envía la capa de aplicación de un sistema pueda ser entendida y utilizada por la capa de aplicación de otro, estableciendo el contexto sintáctico del diálogo.

Su tarea principal es aislar a las capas inferiores del formato de los datos de las aplicaciones específicas, transformando los formatos particulares (ASCII, EBCDIC, etc.) en un formato común de red, entendible por todos los sistemas y apto para ser enviado por red.

Para cumplir estas funciones, la capa de presentación realiza las siguientes operaciones:

- Traducir entre varios formatos de datos utilizando un formato común, estableciendo la sintaxis y la semántica de la información transmitida. Para ello convierte los datos desde el formato local al estándar de red y viceversa.
- Definir la estructura de los datos a transmitir. Por ejemplo, en el caso de un acceso a base de datos, definir el orden de transmisión y la estructura de los registros.
- Definir el código a usar para representar una cadena de caracteres (ASCII, EBCDIC, etc.).
- Dar formato a la información para visualizarla o imprimirla. Comprimir los datos si es necesario.
- Aplicar a los datos procesos criptográficos cuando sea necesario.

Esta capa por lo tanto puede manejar archivos de: texto, datos, gráficos, imágenes, sonido y video para enviarlos como flujo de bits a la siguiente capa.

6 SESIÓN (CAPA 5)

La capa de sesión proporciona sus servicios a la capa de presentación, proporcionando el medio necesario para que las entidades de presentación de dos host que se están comunicando por red organicen y sincronicen su diálogo y procedan al intercambio de datos.

Sus principales funciones son:

- Establecer, administrar y finalizar las sesiones entre dos hosts (máquinas en red) que se están comunicando.

- Si por algún motivo una sesión falla por cualquier causa ajena al usuario, restaurar la sesión a partir de un punto seguro y sin pérdida de datos o, si esto no es posible, terminar la sesión de una manera ordenada, chequeando y recuperando todas sus funciones, evitando así problemas en sistemas transaccionales.

- Sincronizar el diálogo entre las capas de presentación de los dos hosts y administrar su intercambio de datos, estableciendo las reglas o protocolos para el dialogo entre máquinas, regulando quien habla y por cuanto tiempo.

- Conseguir una transferencia de datos eficiente y un registro de excepciones acerca de los problemas de la capa de sesión, presentación y aplicación.

- Manejar **tokens**. los tokens son objetos abstractos y únicos que se usan para controlar las acciones de los participantes en la comunicación, base de ciertos tipos de redes, como Token Ring o FDDI.

- Hacer **check points**, que son puntos de referencia en la transferencia de datos, necesarios para la correcta recuperación de sesiones perdidas.

Como una de las funciones de esta capa es establecer, administrar y terminar una sesión entre aplicaciones, esta tarea normalmente pude comenzar con una solicitud de una máquina a otra para realizar una consulta de SQL (Structured Query Language - Lenguaje de Búsqueda Estructurado) y coordinará las comunicaciones hasta la terminación de la misma.

Otras aplicaciones que pude llegar a administrar son:

RPC (Remote Procedure Call - Llamada de Procedimiento Remoto) es un protocolo que permite a un programa residente en una máquina ser ejecutado por otra en una ubicación remota

NFS (Network File System - Sistema de Archivos de Red) esta es una aplicación cliente servidor diseñada para poder acceder desde una red a archivos compartidos en una máquina o servidor

Para ver como se desarrolla el comportamiento de estas últimas capas diremos que equipo desea comunicarse a través de una aplicación, por ello iniciará una solicitud que será enviada a la capa de presentación para que la información sea convertida de formato, a su vez esta la envía a la capa de sesión que solicitará un inicio de comunicación y la administrará hasta su terminación. Supongamos ahora que deseo abrir otra página de Internet en una nueva ventana, esto implicaría que necesito realizar una nueva conexión en forma independiente a la que estoy llevando a cabo en este momento, aquí es cuando esta capa se dice que administra ya que inicia una nueva sesión o las que se les soliciten hasta su finalización.

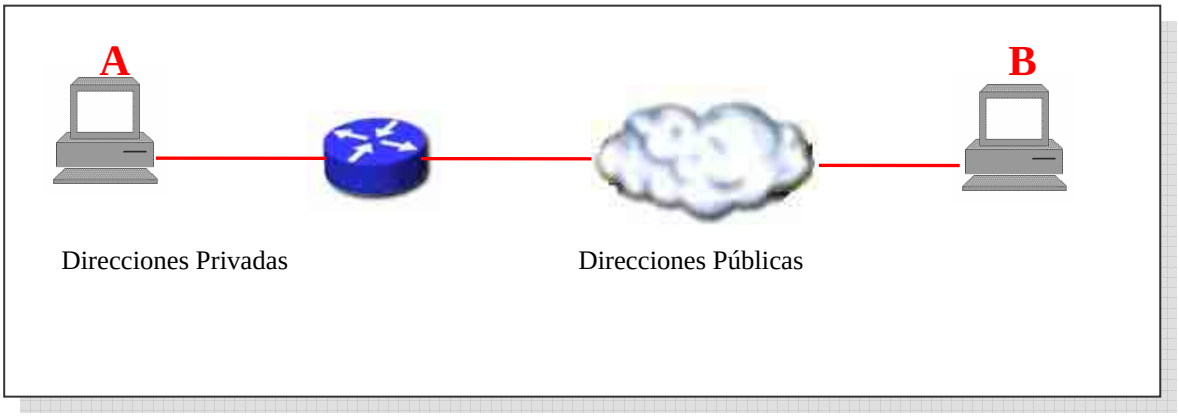
El próximo paso ya lo imaginamos y es el enviar esta información a la capa inferior de transporte para que decida que protocolo se utilizará según la información.

Sabiendo que tipo de comunicación se esta solicitando, primero le asignará un número de puerto de origen que pertenezca al rango de usuarios finales, que podría ser 2020, después asignará un número de puerto destino en nuestro caso será el 80 que es el predeterminado de la aplicación, luego con

la ayuda de los campos específicos para el seguimiento se mantendrá el control para asegurar la integridad de los paquetes enviados.

7 COMUNICACIÓN ENTRE REDES PÚBLICAS Y PRIVADAS.

Hasta ahora estuvimos estudiando la forma de interconectar redes o subredes en el ámbito de las direcciones públicas, a continuación analizaremos que sucede cuando desde una red privada se intenta alcanzar un host ubicado en una red pública.



Supongamos que nos estamos en la máquina ***A*** y a través de un ***router*** queremos comunicarnos con la maquina ***B*** que se encuentra en una red pública.

El proceso es sencillo y podemos dividirlo en dos partes para analizarlo, primero como llegar al host de destino ***B*** desde nuestro host ***A*** y luego el camino inverso, desde el host ***B*** hacia el ***A.***

Situados en el host ***A,*** primero se armará un datagrama con la dirección IP de destino (host ***B***) y origen (host ***A***), este será colocado dentro un frame y le será enviado a nuestro ***router***, este aplicará la máscara, determinará que esta red no le pertenece y utilizará la ruta default gateway que se le halla definido para salir (IP asignada por nuestro ISP). De esta forma ya nos encontramos dentro de las redes públicas y sólo nos resta que los routers de Internet encuentren la mejor ruta para acceder a nuestro host ***B***.

El camino inverso es iniciado por el host ***B*** el cual tiene la tarea de armar un datagrama con las direcciones IP de destino/origen y colocarlas en un frame para enviarlo de regreso al Host ***A***, pero debemos recordar que la dirección de destino ahora es una privada y no pública, por lo tanto el frame enviado de regreso podrá salir del host, pero los routers de Internet aplicarán máscara de red y descartaran el frame por no ser una red conocida.

Esto nos lleva a que el router no es el medio adecuado para establecer una comunicación entre redes privadas y publicas, recordemos que nos encontramos trabajando en la capa red del modelo OSI, y la misma no tiene implementada una solución para este problema.

La solución sólo podrá ser dada por servicios que oficien de intermediarios entre las redes privadas y públicas, tales como NAT y Proxy que trabajan en las capas superiores del modelo OSI (4 a 7).

8 NAT Y PROXY

La utilización de routers para enlazar redes públicas y privadas no es el medio adecuado para hacerlo, el motivo es que trabaja en la capa de 3 y sólo tiene conocimiento sobre redes públicas y no del tipo privadas. Por lo tanto cuando se quiere enviar un paquete desde un host ubicado en la red pública, este será descartado por el primer router por no conocer a la red privada que tiene como destino.

La solución a este problema es colocar un intermediario que entienda como debe enviar y recibir los paquetes entre ambas redes, esta tarea puede ser llevada a cabo por dos productos que son similares, pero con algunas diferencias en cuanto al modo de llevarlas a cabo.

Estos productos son **NAT (*Network Address Translation*** – Traductor de Direcciones de Red) y **PROXY**, este último no es una sigla y se lo puede traducir como delegado (que actúa en nombre de), de los cuales a continuación analizaremos su funcionamiento, ventajas y limitaciones.

Antes de comenzar debemos comprender que estos productos son servicios que pueden ser brindados tanto por un hardware específico, como por software, y que las capacidades que describiremos son sólo las básicas, ya que cada fabricante le podrá agregar otras funcionalidades, logrando así una ventaja comercial.

8.1 NAT

El servicio de NAT se utiliza a diario y sin embargo no es tan conocido, la razón es por que está integrado dentro de otro producto que conocemos como ***ICS*** (*Internet Connection Sharing* - Conexión Compartida de Internet) que se encuentra integrado desde hace tiempo en los SO de Microsoft Windows 98 SE / Me / 2000 y XP, junto a NAT también hay otros servicios como DHCP que son utilizados para simplificar la configuración de las interfaces de red en los host pertenecientes a pequeñas redes domesticas.

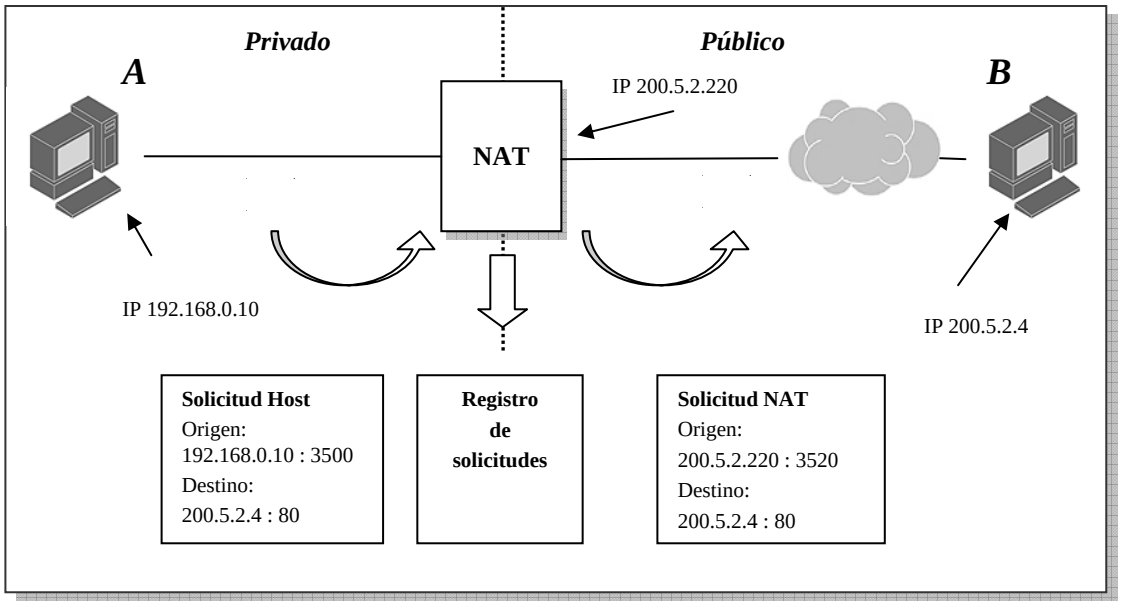
Para explicar el funcionamiento de NAT podemos valernos de un ejemplo muy común, analizaremos su comportamiento de una pequeña red privada que necesita navegar por Internet, o a lo que nosotros denominamos redes públicas.

En la siguiente figura tenemos un host ***A*** que se encuentra en una red privada, un servicio de NAT que posee un acceso a redes y el host ***B*** del cual conocemos su dirección IP en la red pública.

- El primer paso consiste en que el host ***A*** envía a la red un frame, el cual transporta un datagrama IP (paquete) que contiene la dirección de destino IP 200.5.2.4 perteneciente al host ***B***, y la de origen 192.168.0.10.
- Este frame es captado por NAT, que ahora es el encargado de las comunicaciones con las redes públicas, y realiza las siguientes operaciones.

- Abre el frame, y en el datagrama reemplaza la dirección de origen perteneciente al host ***A***, reemplazándola por la propia 200.5.2.220 (una dirección publica).
- Otro cambio que realiza es el del puerto de salida.
- NAT en este preciso momento también abre un registro de esta solicitud y de todas aquellas que se realicen desde la red privada por cualquier host, tomando nota de la direcciones IP de origen y la IP solicitada.

- Por último envía a la red el frame ya modificado, asegurándose de esta forma el retorno de la información.



El procedimiento de retorno es muy sencillo, ahora el host ***B*** envía un frame respondiendo, a la dirección IP del NAT (200.5.2.220).

- NAT recibe el frame reconociéndolo como propio, debe abrirlo para constatar la dirección IP y consultar en su registro quien fue el solicitante.
- Finalmente reemplaza la dirección IP de destino del datagrama 200.5.2.220 por la del host ***A*** (192.168.0.10) quien fue el solicitante.

Como síntesis de este procedimiento, podemos decir que el servicio de NAT se dedica a pasar los frames del lado privado hacia el lado público reemplazando las IP en los datagramas, siendo un servicio invisible tanto para los usuarios, como para las aplicaciones.

8.2 PROXY

El servicio de proxy es muy difundido y tiene en principio la misma funcionalidad que NAT enlazar redes privadas y públicas, este es un delegado que se posiciona entre ambas redes realizando el reemplazo de las IP del solicitante por la propia, si bien esta tarea es aparentemente la misma, la forma en que la lleva a cabo es totalmente distinta ya que realiza otras tareas. Un proxy interceptará todas las peticiones de servicio ya sea que provengan del lado privado o el público y actuará como representante evitando así una consulta directa entre ambas redes, aquí debemos recordar que NAT solo se dedica a traspasar los paquetes entre ambas redes.

Normalmente estamos acostumbrados a que el servicio de proxy o servidor proxy, viene con una gran variedad de funcionalidades extras, aquí debemos remarcar que un servidor proxy es solo un delegado y opera en forma individual para cada tipo de protocolo, por ejemplo HTTP, FTP, POP, SMTP, etc.

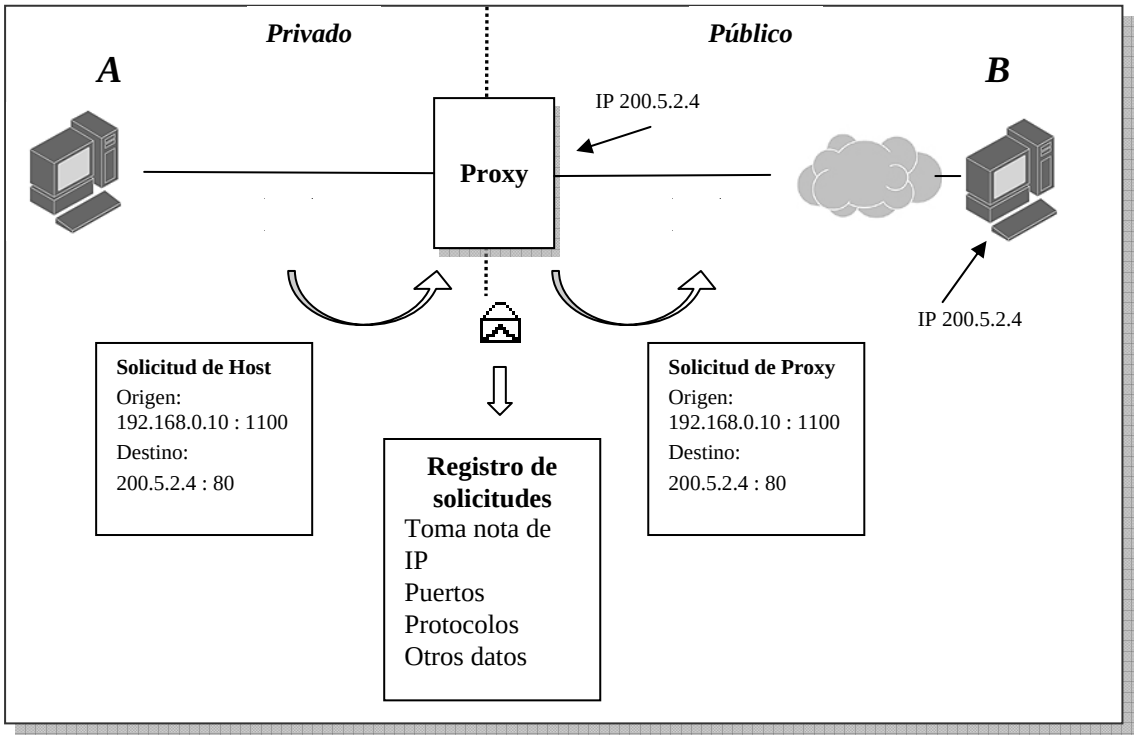
Otras funcionalidades que se encuentran son consideradas como estándar, pero en realidad son agregadas por el fabricante para aumentar las prestaciones de su producto y así lograr una posición más competitiva en el mercado. Estas extras actualmente son el servicio de memoria cache para páginas web, firewall, servidores de correo interno, etc.

A continuación llevaremos acabo un análisis sobre el funcionamiento de un proxy en su forma más pura y luego abordaremos algunas de sus ventajas debido a su forma de trabajo.

Para realizar nuestro trabajo nos apoyaremos en la próxima figura, que tiene las mismas características de la anterior.

Al comienzo dijimos que la tarea primaria que cumple es la misma, y que la diferencia radica en la forma de realizarla, cuando el proxy recibe el frame enviado por el host ***A*** lo abre y no sólo cambia la dirección IP de origen en el datagrama, también realiza las siguientes operaciones:

- Cada solicitud de un host es una sesión de proxy independiente, la cual será auditada desde el comienzo hasta su finalización.
- Proxy lleva un registro de los números de puertos de origen y destino que se utilizan en capas superiores.
- También se implementa un servicio que atenderá a cada aplicación de red, por su puerto de comunicación predeterminado.
- Los números de puertos predeterminados para atender cada servicio podrán ser cambiados por otros, según las necesidades. Un ejemplo de esto sería si existiese un servidor web en la misma máquina que el servidor proxy.



Ampliando lo antedicho en las capacidades del proxy, cuando nos referimos a las aplicaciones de red, ellas son Internet o Correo, sus clientes son el Internet Explorer (IE) - Outlook Express, que utilizan los protocolos HTTP - POP3 - SMPT, y estos a su vez utilizan los puertos 80 - 25 - 110 que son los predeterminados para atender las solicitudes desde los servidores de esta aplicaciones.

Estas nuevas características nos dan la posibilidad de retirar o agregar los llamados servicios en el servidor proxy, esto quiere decir que si lo deseamos podríamos dejar sin servicio de descarga de archivos a toda la red con sólo no atender al protocolo FTP que atiende todas las solicitudes que se dirigen hacia el puerto 21.

También es posible cambiar los números de dichos puertos, esto parece muy extraño pero la razón para esto es los servidores proxy (en versión software), pueden llegar a coexistir dentro de una máquina que tenga implementados servidores web o de correo, por ejemplo al tener un servidor Web corriendo sabemos que este atiende en el puerto 80, pero nuestro servidor proxy también intercepta todas las solicitudes de http realizada por el IE y esta será interceptada por nuestro proxy

La solución a este problema se puede concretar en dos pasos, la primera acción se llevaría a cabo en nuestro servidor proxy donde debemos especificar un nuevo numero de puerto al servicio WWW, en este caso particular podemos utilizar el 8080. El segundo paso lo realiza en el cliente (IE), al cual le debemos informar que a partir de ahora el nuevo número de puerto por donde tendrá que salir es el 8080, siendo este el que definimos en el servicio WWW del proxy.

Esta no sería la única aplicación que se le puede encontrar, y por eso es que se asocian a los servidores proxy con una gran cantidad aplicaciones extras que no son nativas del mismo, pero que sin embargo pueden llegar a cumplir gracias a estas funcionalidades.

Estas funciones extras van desde crear una memoria cache en disco para guardar páginas web, para ser utilizadas posteriormente ante una consulta que se realice al mismo sitio y así ahorrar tiempo de navegación, disminuir el tiempo de acceso a los contenidos solicitados por del usuario.

Otras funciones son directamente agregados que nada tienen ver con un servidor Proxy, por ejemplo DHCP, DNS, servidores de correo internos, etc.

Una aplicación que si puede llevar a cabo un servidor proxy y que desarrollaremos en futuras clases es la de Firewall.

NOTAS

CUESTIONARIO CAPITULO 13

1.- ¿Cual es el protocolo de transporte que no utiliza mecanismos para control de conexión?

2.- ¿Cual es el propósito del protocolo de transporte UDP?

3.- ¿Cual es la capa del modelo OSI que introduce al Puerto y cual es fundamento?

4.- Si tengo tres archivos descargándose simultáneamente desde Internet. ¿Quien se encuentra administrado la misma?

5.- ¿Si tengo un host que sale a Internet a través de un NAT, las aplicaciones que tareas deben cumplir para poder salir?