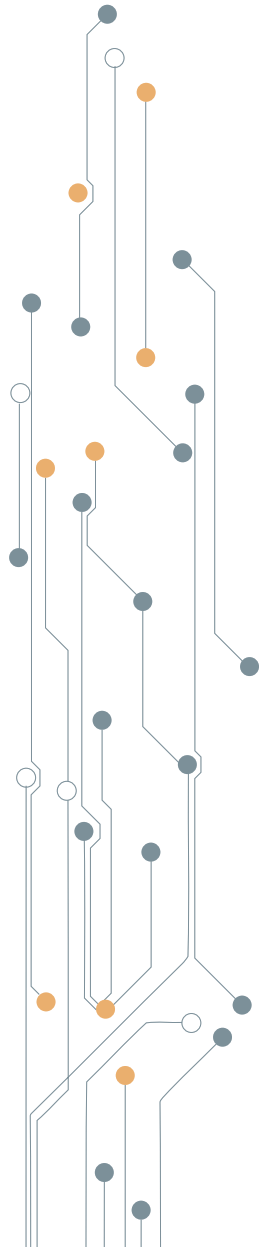
Características de una red

Telefónica EDUCACIÓN DIGITAL

Índice



1 | Introducción	3
2 | Nivel de alcance	4
3 | Nivel de red	5
3.1 | Direccionamiento	6
3.2 | Internet Protocol: IP	6
3.3 | ARP	13
4 | Nivel de transporte	15
4.1 | UDP	15
4.2 | TCP	18
5 | NAT	24
6 | Nivel de aplicación	32
6.1 | HTTP	32
6.2 | DNS	34
6.3 | SSH	39
7 | Ejemplo final	41

1. Introducción

- Una red debe ser rápida, fiable, de bajo coste, escalable, disponible y sobre todo segura.
- En una red estructurada en capas, cada nivel ofrece un determinado servicio o tarea a la capa superior, sin enviar información acerca de la tarea de la que se ha encargado ella. Entre los diferentes niveles se establece un interfaz.
- Cada capa en un dispositivo “habla” con su correspondiente en otra capa. Las reglas del idioma en el que “hablan” constituyen lo que llamamos un protocolo.
- No se transmiten físicamente datos entre niveles gemelos. La transferencia se realiza en cada dispositivo, entre capas adyacentes, mediante las interfaces ente capas.

2. Nivel de enlace

El nivel de enlace o capa 2, como estudiamos anteriormente, es la capa encargada de transformar el sistema de transmisión resultante del nivel físico, en una línea libre de errores de transmisión.

El nivel de enlace trabaja con una unidad de envío de datos denominada trama. Dicha trama es una serie sucesiva de bits, con estructura cíclica, a través de los que se transporta información.

Su principal cometido es formar las tramas que posteriormente pasarán al nivel físico a partir de los datos recibidos del nivel de red, así como reconocer dichas tramas e identificar los datos de las tramas recibidas del nivel físico y que a continuación pasará al nivel de red.

Podemos decir que las funciones básicas del nivel de enlace son las siguientes:

- Control de acceso al medio.
- Detección de errores.
- Aceptación de paquetes del nivel 3, empaquetándolos en tramas.

La capa de enlace separa de forma eficiente las transiciones existentes cuando un paquete es reenviado hasta las capas superiores. La capa de enlace, en definitiva, coge los paquetes de la capa superior y los guía a un protocolo de las mismas propiedades. No necesita conocer qué medios usa la comunicación.

El nivel de enlace puede subdividirse en dos categorías:

- **MAC (Control de acceso al medio).** Es la capa inferior y su principal función es definir los procesos para acceder al medio de transmisión compartido por varias máquinas. Proporciona el direccionamiento del nivel de enlace y la limitación de los datos de acuerdo a los requisitos del medio.
- **LLC (Control de enlace lógico).** Es la capa superior y la responsable del control del enlace lógico. Coloca en la trama datos identificativos respecto a la capa de red utilizada para la trama. Ofrece al nivel de red un servicio de transmisión de información en forma de bits entre máquinas adyacentes, manejando el control de flujo, la gestión de tramas y el control de errores.

La división de la capa de enlace en subniveles hace que una trama definida por el nivel superior pueda acceder a diferentes medios que el nivel inferior define.

3. Nivel de red

El nivel de red corresponde a la capa 3 del modelo de OSI y tiene como objetivo el intercambio de datos entre dos dispositivos en una determinada red. Para poder llevar a cabo ese intercambio, el nivel de red se basa en cuatro pilares básicos:

- **Direccionamiento:** configuración de los dispositivos con un elemento identificativo, la dirección IP.
- **Encapsulado:** la unidad de datos del nivel de red es el paquete, que está formado por la información que proviene de la capa de transporte (PDU), más un encabezado IP al que se añade información como la IP de origen y destino.
- **Rutas:** el nivel de red se encarga de proporcionar la ruta que debe seguir el paquete hasta llegar a su destino. Para ello se sirve de routers, que permiten al paquete cambiar de red.
- **Desencapsulado:** cuando el paquete llega a su destinatario, la máquina deberá ser capaz de desencapsular el paquete para comprobar que la IP destino corresponde a la suya. Una vez que comprueba esto, elimina la cabecera consiguiendo el PDU, la unidad de nivel de transporte, a la que entregará para continuar el proceso de envío.

3.1 | Direccionamiento

El direccionamiento es una tarea básica en el nivel de red, ya que hace posible que la conexión y la transmisión de datos entre dos máquinas diferentes de la misma o diferentes redes sea posible. Mediante la asignación de direcciones únicas a nivel de red para las máquinas podremos facilitar la tarea del encaminamiento, incluyendo información acerca del destinatario y de cómo hacerle llegar el paquete que queremos enviarle. El direccionamiento debe cubrir aspectos como el cambio de localización de una determinada máquina, en la que debe cambiar su dirección y consecuentemente, las entradas de las tablas de enrutamientos de los diferentes routers.

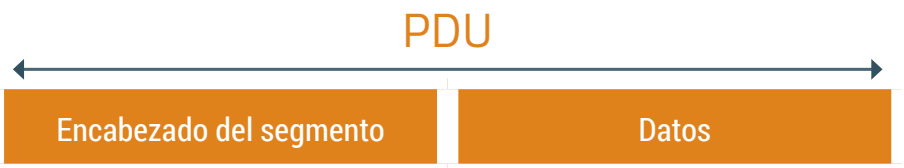
3.2 | Internet Protocol: IP (IPv4 e IPv6)

El protocolo IP es la prestación del nivel de red desarrollado por los protocolos TCP/IP. Gracias a este protocolo un paquete puede ser enviado desde una máquina a otra a través de una red interconectada. Las principales características son:

- **No conexión:** no realiza una conexión previa con la máquina destino antes de proceder a enviar el paquete. Tampoco necesita campos adicionales en el encabezado del PDU para continuar una comunicación ya establecida. Gracias a esto la sobrecarga del protocolo se reduce considerablemente.
- **No fiable:** no se garantiza la llegada del paquete, no tiene la capacidad para la administración de paquetes que no han sido entregados ni la recuperación de los mismos (las capas superiores se encargan de esto). Esto es debido a que los paquetes en envían con datos acerca del proceso de entrega, pero no con datos procesables por el emisor para el conocimiento de la correcta entrega. No existen por tanto los acuses de recibo ni control de errores ni retransmisión de paquetes.
- **Independiente del medio:** no depende del medio por el que los datos son transportados.

Encapsulación de IP

Como se acaba de indicar, el protocolo IP añade una cabecera IP al segmento que proviene de la capa de transporte, con el objetivo que aportar información del destino del paquete. En la capa de transporte se originaba un PDU similar al que se muestra en la siguiente figura:



ENCAPSULACIÓN DE LA CAPA DE TRANSPORTE

FIGURA 2.2.2.1 ENCAPSULACIÓN CAPA DE TRANSPORTE

En la siguiente figura, se muestra el siguiente proceso de encapsulado, correspondiente a la unidad de la capa de red:

ENCAPSULACIÓN DE LA CAPA DE RED

FIGURA 2.2.2.2 ENCAPSULACIÓN CAPA DE RED

El nivel de red añade un encabezado para que el enrutamiento de los paquetes a través de redes interconectadas sea posible, y así el paquete pueda llegar a su destino. En las redes basadas en TCP/IP, la unidad de datos se conoce como paquete IP.

El proceso de encapsulado a capas tiene la ventaja de proporcionar independencia entre capas, ya que la adición de diferentes cabeceras hace que no interfieran entre ellas. Los routers pueden implementar distintos protocolos de capa de red, el enrutamiento que llevan a cabo estos dispositivos sólo tiene en cuenta el encabezado del paquete encapsulado sin modificar la unidad original proporcionado por la capa anterior (PDU).

IPv4

Internet se basa en su gran mayoría en IPV4. Este protocolo surgió en 1983 y a día de hoy sigue siendo el protocolo del nivel de red más utilizado.

Los paquetes IPV4 están divididos en dos partes:

- Encabezado IP: Detalla las especificaciones del paquete.
- Datos: Corresponden a la información del segmento del nivel 4 y los datos en sí.

El encabezado a su vez tiene diferentes campos:

- **Versión:** cuatro bits, que en IPv4 siempre son 0100.
- **DS (Servicios diferenciados):** ocho bits que determinan la prioridad del paquete. Los seis primeros corresponden al DSCHP(Punto de código de servicios diferenciado) y los dos restantes corresponden al ECN (Notificación Explícita de congestión).
- **TTL (Tiempo de vida):** ocho bits que definen el tiempo de vida útil del paquete. El valor inicial es establecido por la máquina emisora y va disminuyendo en cada salto. Este valor no puede llegar a cero, ya que en tal caso el paquete será descartado.
- **Protocolo:** ocho bits que especifican la tipología del paquete (TCP, UDP...).
- **IP origen:** treinta y dos bits que indican la dirección origen del paquete.
- **IP Destino:** treinta y dos bits que indican la dirección destino del paquete.
- **IHL (Longitud del encabezado de Internet):** cuatro bits que especifican en binario el número de palabras de 32 bits del encabezado.
- **Longitud total:** dieciséis bits que especifican el tamaño total del paquete.
- **Cheksum:** dieciséis bits destinados a la verificación de errores (El valor calculado antes de enviar el paquete y al recibirlo en el destino debe ser el mismo).
- **Identificación:** dieciséis bits que indica el número de fragmento en caso de que un paquete haya tenido que fragmentarse para ser enviado.

- **Indicadores:** tres bits que especifican el modo de fragmentación del paquete, en el caso de que haya habido fragmentación para enviarse.
- **Desplazamiento de fragmentos:** trece bits que especifican el orden en el que deben ordenarse los distintos fragmentos del paquete, en el caso de que haya habido fragmentación para enviarse.

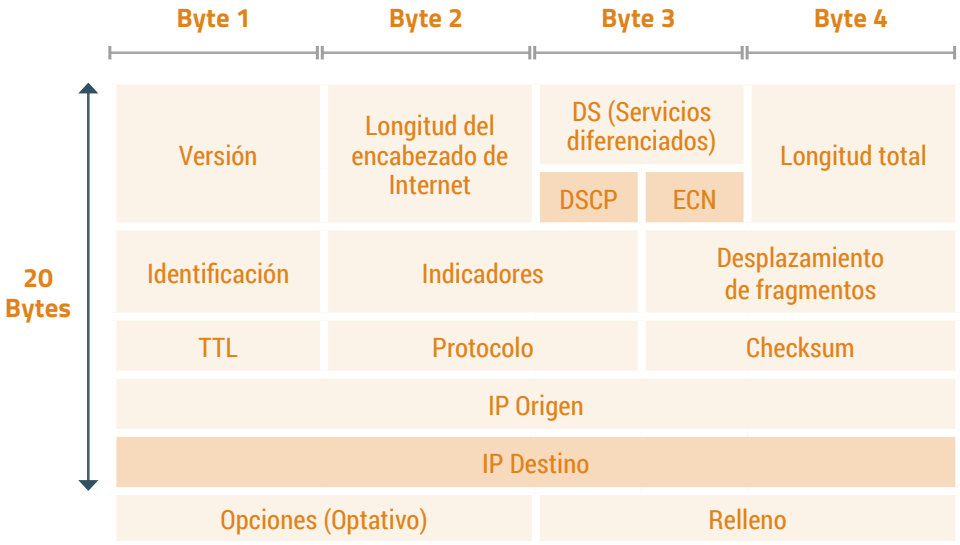


FIGURA 2.2.2.3 PAQUETE IPV4

Pese a ser el protocolo de red más utilizado IPV4 tiene varias limitaciones importantes:

- **Agotamiento de direcciones IP:** la cantidad de direcciones es limitada, y aunque dicha cantidad es muy elevada (4.000 millones) el incremento exponencial a lo largo de los últimos años de dispositivos con IP está haciendo que el número de direcciones necesarias sea cada vez mayor.
- **Dificultades en las tablas de enrutamiento:** como ya se ha estudiado los routers cuentan con tablas de enrutamiento en las que almacenan rutas, indicando la dirección del siguiente salto. Como cada vez hay más direcciones IPv4, el número de rutas entre dispositivos también aumenta, aumentando también los recursos consumidos por estas rutas en cuanto a memoria y procesador.
- **No conectividad entre extremos finales:** debido al uso creciente de NAT, una solución que hace que varios dispositivos compartan una IP pública a través de un router NAT que forma una subred, cada vez puede ser más problemático el uso de dispositivos end-to-end que desconocen las direcciones finales de dispositivos que se encuentran tras un router NAT.

Pese a ser el protocolo de red más utilizado IPV4 tiene varias limitaciones importantes:

- **Agotamiento de direcciones IP:** la cantidad de direcciones es limitada, y aunque dicha cantidad es muy elevada (4.000 millones) el incremento exponencial a lo largo de los últimos años de dispositivos con IP está haciendo que el número de direcciones necesarias sea cada vez mayor.
- **Dificultades en las tablas de enrutamiento:** como ya se ha estudiado los routers cuentan con tablas de enrutamiento en las que almacenan rutas, indicando la dirección del siguiente salto. Como cada vez hay más direcciones IPv4, el número de rutas entre dispositivos también aumenta, aumentando también los recursos consumidos por estas rutas en cuanto a memoria y procesador.
- **No conectividad entre extremos finales:** debido al uso creciente de NAT, una solución que hace que varios dispositivos compartan una IP pública a través de un router NAT que forma una subred, cada vez puede ser más problemático el uso de dispositivos end-to-end que desconocen las direcciones finales de dispositivos que se encuentran tras un router NAT.

IPv6

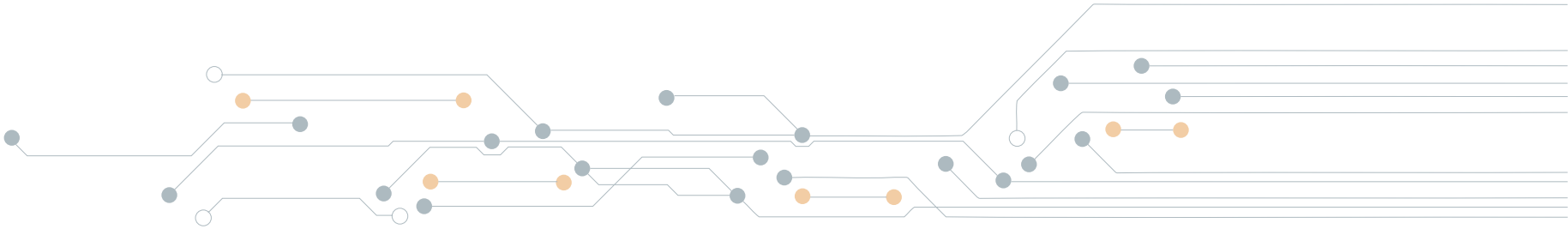
Debido a la cantidad de limitaciones que comenzaron a surgir con IPv4, pronto se empezó a pensar en posibles soluciones. De esta forma, a finales de los 90 surgió IPv6, con las siguientes ventajas:

- **Mayor número de direcciones:** como en IPv4 las direcciones son de 32 bits y en IPv6 son de 128 el número de direcciones disponibles es exponencialmente proporcional. Con IPv4 las direcciones asignables eran alrededor de 3700 millones, mientras que en IPv6 la cifra asciende a 340 sextillones de direcciones asignables).
- **No necesidad de soluciones como NAT:** debido a que el número de direcciones IP disponibles es mucho mayor no es necesario el uso de soluciones como NAT (con los consecuentes problemas que puede provocar en aplicaciones end-to-end).
- **Mayor seguridad:** IPv6 tiene un sistema de autenticación y privacidad con el que IPv4 no contaba.

El diseño del encabezado también es diferente al de IPv4, siendo más simple en la nueva versión. El nuevo encabezado está formado por 40 octetos (frente a los 20 de IPv4, pesa más porque las direcciones tienen más longitud) y 8 campos destinados al encabezado.

Algunos de los campos de IPv4 permanecen, otras no son usadas y en algunas de ellas se ha cambiado el orden. El nuevo encabezado proporciona una mayor eficacia en el enrutamiento, mayor simplicidad y prescinde del uso del cheksum. Los nuevos campos del encabezado son los siguientes:

- **Versión:** cuatro bits que identifican la versión, en IPv6 siempre serán 0110.
- **Tipo de tráfico:** ocho bits que son equivalentes al DS de IPv4. Los seis primeros son para la clasificación del paquete y los dos últimos para la gestión del tráfico.
- **Identificador de flujo:** veinte bits que tienen el objetivo de informar a los routers que deben seguir la misma routa para todo el flujo de paquetes.
- **Longitud de contenido:** dieciséis bits que especifican el tamaño total del paquete.



- **Siguiente encabezado:** ocho bits que indican la clase de contenido del paquete
- **Límite de saltos:** ocho bits que tienen la misma función que el TTL de IPv4.
- **Dirección de origen:** ciento veinte y ocho bits que especifican la IP origen.
- **Dirección de destino:** ciento veinte y ocho bits que especifican la IP destino.

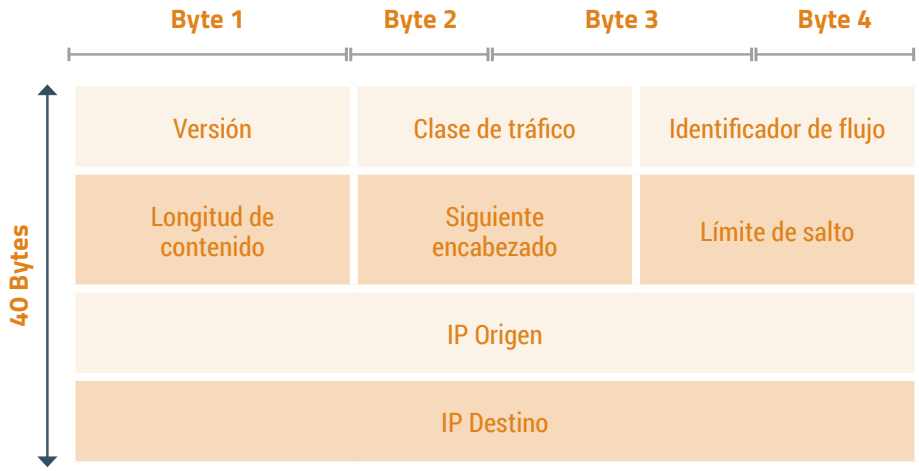
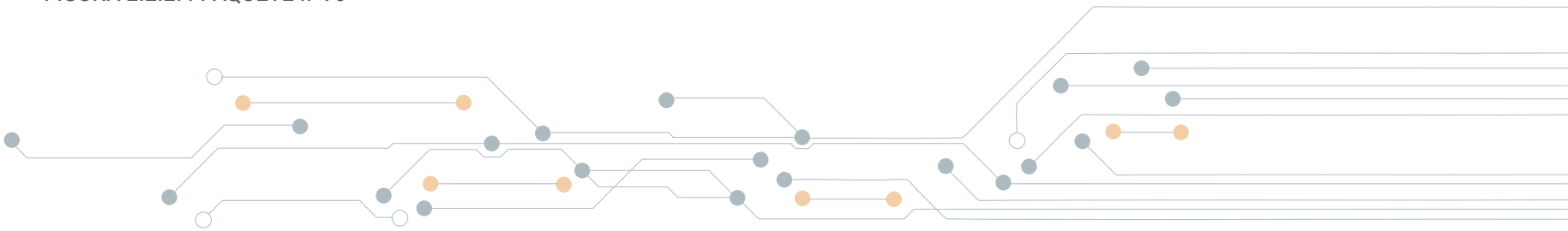


FIGURA 2.2.2.4 PAQUETE IPV6



3.3 | ARP

ARP (Address Resolution Protocol) es el protocolo de resolución de direcciones mediante la cual es posible transformar una dirección IP en una dirección Ethernet. Para ello, cuando la capa de red, con el protocolo IP, va a enviar un paquete con una dirección IP específica tiene en cuenta lo siguiente:

- ¿La dirección de destino pertenece a la misma subred? Si la respuesta es sí, esa máquina es la máquina a la que hay que enviar directamente la trama Ethernet que contenga el paquete. Si la respuesta es no, entra en juego la tabla de encaminamiento, que proporciona la dirección IP del siguiente intermediario, que es el encaminador al que hay que enviar la trama Ethernet que contenga el paquete.
- En ambos casos, sólo hay conocimiento de la IP del siguiente salto, mientras que en la trama Ethernet es necesario saber la dirección Ethernet del destino final.

Para saber la dirección Ethernet de un a máquina que se encuentra en la misma subred a partir de la dirección IP, la máquina debe realizar el siguiente proceso:

- La máquina envía una trama Ethernet en modo broadcast que en realidad es una solicitud ARP. Esta solicitud contiene la dirección IP correspondiente.
- La máquina que reciba la solicitud ARP e identifique que la dirección IP por la que se pregunta es la suya responderá con una trama Ethernet con destino a la máquina que originó la petición ARP. La trama de vuelta será una respuesta ARP con la dirección Ethernet solicitada.

Cada máquina tiene una caché de direcciones IP con sus respectivas relaciones con las direcciones Ethernet, basándose en las solicitudes que va realizando.

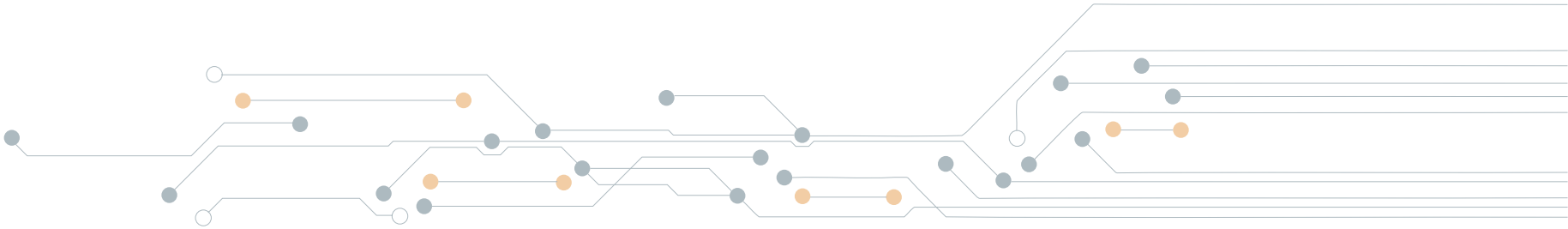
El formato para usar ARP con IP es el siguiente:

No obstante, ARP puede utilizarse también con otros protocolos de red (no sólo IP) y dependiendo del protocolo el formato variará.

Debemos tener en cuenta que el paquete de ARP viaja en el interior de una trama Ethernet. En una solicitud, los datos correspondientes al origen contienen la información de la máquina que pregunta y en los correspondientes a destino, el campo “Ethernet Destino” irá en blanco, ya que esa información se completará en el paquete respuesta. De esta forma, en el paquete respuesta, la máquina a la que se ha preguntado la dirección Ethernet cambiará los datos de origen por los del destino, completando la dirección Ethernet pedida (la suya propia). Como las solicitudes ARP son enviadas por medio de broadcast, cuando una máquina reciba una actualizará siempre la entrada “IP Origen” con su correspondiente “Eth. Or”.

NOTA

Una máquina tiene la opción de mandar una solicitud ARP para pedir su propia dirección IP con el objetivo de actualizar su dirección IP y Ethernet en las tablas de las demás máquinas. También puede hacerlo para evitar e identificar posibles direcciones IP duplicadas.



4. Nivel de transporte

La capa de transporte se encarga de iniciar la comunicación entre dos dispositivos o aplicaciones y de transmitir información entre ellos. Además, la capa de transporte es el enlace entre la capa de aplicación y las capas que se encuentran por debajo y tienen como misión la transmisión a través de la red de todos los paquetes de información.
La función principal de la capa de transporte es proporcionar un sistema para enviar información a través de la red, de forma que posteriormente se puedan volver a anexionar de forma correcta cuando hayan llegado al receptor. Esta capa soporta la segmentación de los datos y la capacidad para volver a unirse en los diferentes streams de comunicaciones, mediante dos protocolos muy diferentes: UDP y TCP.

4.1 | UDP

UDP responde a las siglas "User Datagram Protocol" y es un protocolo simple implementación por el nivel de transporte, que está orientado a datagramas. Se basa en dos puntos principales: NO es orientado a conexión y NO es fiable. UDP se considera un protocolo de máximo esfuerzo, ya que no hay acuse de recibo que informa que los datos han sido recibidos, y ofrece la misma segmentación y anexión de datos que TCP, sin su fiabilidad y su control de flujo.

UDP se caracteriza por las siguientes funcionalidades:

- **No orientado a conexión:** este protocolo no estable una conexión previa entre el origen y el destino antes del envío de información.
- **No fiable:** UDP no proporciona la seguridad de que los datos se entregan, ya que no dispone de procesos que hagan que se reenvíe información en el caso de que los paquetes no lleguen al destino final.
- **No ordenado:** existen situaciones en la que los paquetes de información llegan al destinatario en un orden diferente al que se han enviado. UDP no posee ningún sistema que permita la ordenación de dichos paquetes.
- **Sin control de flujo:** UDP tampoco dispone de procesos para tener un control de la información que se transmite en cuanto a cantidad. Con esto existe riesgo de saturación, y si el destinatario sufre sobrecarga descartará los datos recibidos por el origen perdiendo la información, ya que, como se acaba de indicar, el origen no retransmite automáticamente los datos enviados.

El proceso de UDP es parecido al servicio postal ordinario en el que no hay seguimiento de envío. Una persona envía una carta a un destinatario, pero no conoce si el receptor recibirá la carta ni recibirá confirmación de envío por parte de la compañía de correos.

Las porciones de comunicación se denominan datagramas y tienen la siguiente estructura:

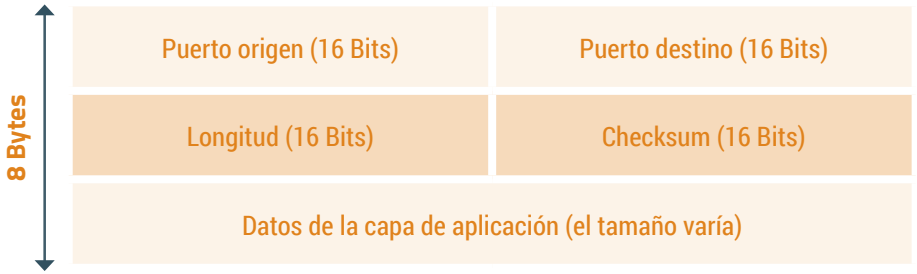


FIGURA 2.3.1. DATAGRAMA UDP

Los datagramas UDP se encapsulan en la parte de datos de un datagrama IP. De esta forma, si la aplicación que use UDP desea transmitir información, generará en cada ocasión un datagrama UDP, que a su vez estará incrustado en un datagrama IP. Si ese datagrama IP tiene mayor tamaño que la unidad de datos de enlace se dividirá en varios datagramas.

El checksum (una suma de chequeo que tiene como propósito detectar cambios en una secuencia de datos entre el inicio y el final de la comunicación) es sobre cabecera y datos, siendo opcional aunque recomendable. Se calcula en base al datagrama UPD y una cabecera que se coloca delante.

Comunicación UDP

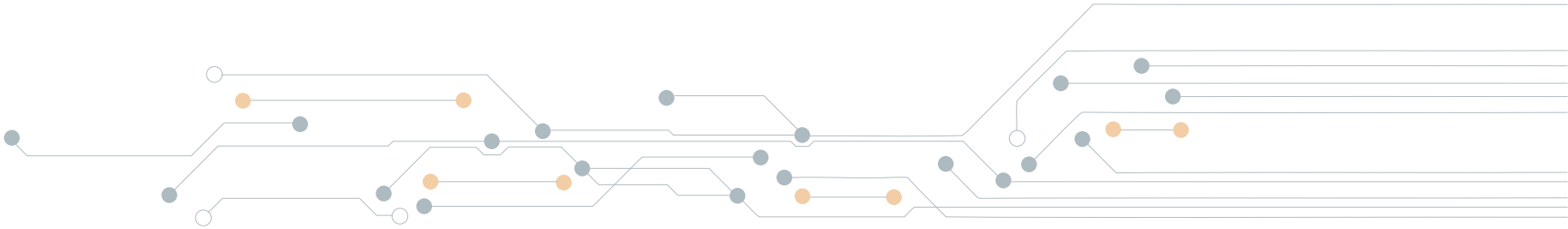
La comunicación cliente/servidor es iniciada por una aplicación que solicita datos de un servidor. Lo que hace el proceso de cliente UDP es asignar aleatoriamente un número de puerto y lo usa como puerto de origen para el diálogo. Los números de puerto de origen seleccionados aleatoriamente proporcionan una mínima seguridad, ya que si siguieran un patrón predecible un atacante podría acceder intentando conectarse al puerto con mayor probabilidad de estar abierto.

Como en UDP no existe comunicación con sesiones, cuando los datos estén listos y los puertos identificados, se generarán los datagramas, que se pasarán a la capa de red para direccionarlos y enviarlos a la red.

Una vez que el cliente también tenga seleccionados los puertos, se añadirán en forma de cabecera en los datagramas de la comunicación.

Ventajas de UDP

Aunque UDP no sea fiable como TCP, la entrega de datos de baja sobrecarga de este protocolo hace de él un protocolo de transporte idóneo en aplicaciones que pueden permitirse una ligera pérdida de datos. Por ejemplo, las aplicaciones de video y voz necesitan transmitir de forma que los datos fluyan muy deprisa y pueden tolerar cierta pérdida de datos con un efecto imperceptible. Para estos casos UDP es un protocolo ideal.



4.2 | TCP

TCP, Transmission Control Protocol, se considera un protocolo fiable, es decir, incluye procesos que garantizan la entrega de información a través del acuse de recibo. En este caso, el proceso es similar a cuando enviamos un paquete y podemos realizar un seguimiento a través de la empresa de mensajería. Las principales características de TCP son las siguientes:

- **Conversaciones orientadas a conexión:** TCP es un protocolo orientado a conexión, en el que se establecen sesiones entre el dispositivo de origen y el destino antes del envío de información. En dicha sesión se prepara la comunicación entre ambos y se negocia la cantidad de tráfico que se puede enviar. La sesión no finaliza hasta que no se haya completado el envío de toda la información.
- **Entrega fiable:** TCP asegura que todos los segmentos de datos lleguen al destino. Si uno de ellos se pierde o se corrompe, el origen retransmitirá el segmento correspondiente.
- **Entrega ordenada:** es posible que los segmentos de datos lleguen desordenados debido a las diferentes rutas que pueden utilizarse y sus diferentes velocidades de transmisión. Esto no supone un problema, ya que en TCP se numeran secuencialmente los segmentos para que en el destinatario se pueda estructurar la información de en el orden correcto.
- **Control de flujo:** TCP puede regular la cantidad de datos que transmite la máquina emisora o reducir la velocidad de envío en caso de que haya sobrecarga, ya que los recursos son limitados. De esta forma se puede evitar la pérdida de segmentos que implican el reenvío correspondiente.

Una vez que TCP ha establecido la conexión es capaz de "monitorizar" el diálogo dentro de esa sesión. Se considera a TCP un protocolo con estado, esto significa que puede realizar un seguimiento de la comunicación. De esta forma, usando TCP la máquina que manda los paquetes espera el acuse de recibo de la máquina destinataria y si dicho recibo no llega procederá a reenviar los paquetes. La sesión con estado empieza al establecer la sesión y acaba con el cierre de sesión.

Cada segmento TCP cuenta con 20 bytes de sobrecarga en la cabecera que encapsula los datos correspondientes a la capa de aplicación. Se trata de un segmento mucho mayor que en UDP, y es que la sobrecarga cuenta con:

- Número de secuencia (32 bits): para la ordenación de datos.
- Número de acuse de recibo (32 bits): para conocer el número de datos que se han recibido.
- Longitud del encabezado (4 bits): para indicar la longitud del encabezado del segmento.
- Bits de control (6 bits): para indicar la función del segmento.
- Tamaño de la ventana (16 bits): para el número de segmentos que se pueden aceptar por vez.
- Checksum (16 bits): para verificar errores en la transmisión.
- Urgente (16 bits): para indicar si los datos son urgentes.

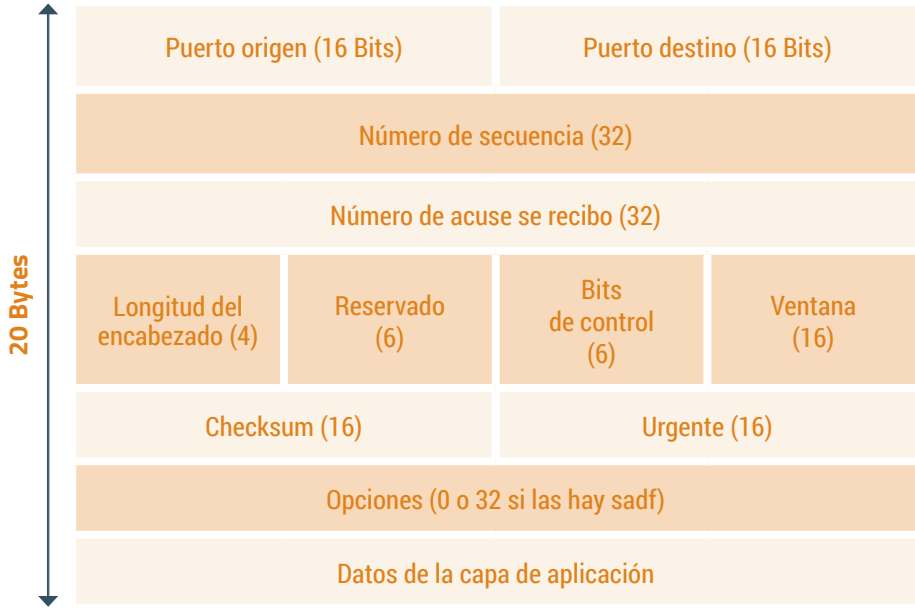


FIGURA 2.3.2.1 SEGMENTO TCP

Establecimiento y finalización de la conexión

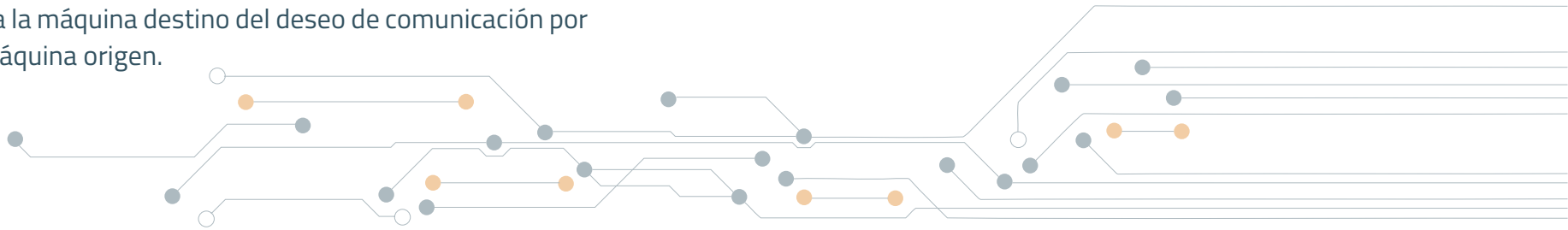
Para comenzar una comunicación orientada a conexión el primer enlace solicita la sincronización mientras que el segundo enlace envía el acuse de recibo sincronizando la conexión en dirección contraria. El tercer segmento es otro acuse de recibo en el que se informa al destino que ambos lados están sincronizados. Cuando las dos máquinas realizan una transmisión de datos previamente, como se ha comentado anteriormente, se establece una conexión. Después de que la transmisión se haya llevado a cabo, se cierran las sesiones y la conexión.

Las máquinas realizan un seguimiento de todos los segmentos enviados e intercambian datos basándose en el encabezado del segmento. TCP es un protocolo full-duplex, en el que la comunicación se pude representar con dos flujos de comunicación unidireccionales, o lo que es lo mismo, las sesiones. Para llevar a cabo la conexión las máquinas llevan a cabo un protocolo de tres vías, que se basa en lo siguiente:

- Establecimiento de la máquina destino en la red.
- Verificación de que la máquina destino tiene un servicio activo y sea capaz de aceptar solicitudes en el puerto en el que la máquina origen está intentando establecer comunicación.
- Información a la máquina destino del deseo de comunicación por parte de la máquina origen.

Antes de proceder a explicar en detalle el proceso de enlace de tres vías se va a proceder a la explicación de diferentes términos que debemos comprender previamente. Estos términos corresponden a campos de 1 bit del interior del segmento y que son necesarios en una comunicación TCP. Son los siguientes:

- URG: Campo que informa de la urgencia de los datos.
- PSH: Función de empuje.
- RST: Campo destinado al reseteo de la comunicación.
- ACK: Acuse de recibo.
- SYN: Sincronización secuencial de números.
- FIN: Fin de datos procedentes del emisor.



Una vez que conocemos los campos anteriores, procedemos a explicar el proceso de enlace de tres vías:

- CTL=Bits de control establecidos en el primer paso, en el encabezado TCP (A envía una solicitud SYN a B).
- Posteriormente B envía una respuesta ACK y
- una solicitud SYN a A y por último A envía una respuesta ACK a B.

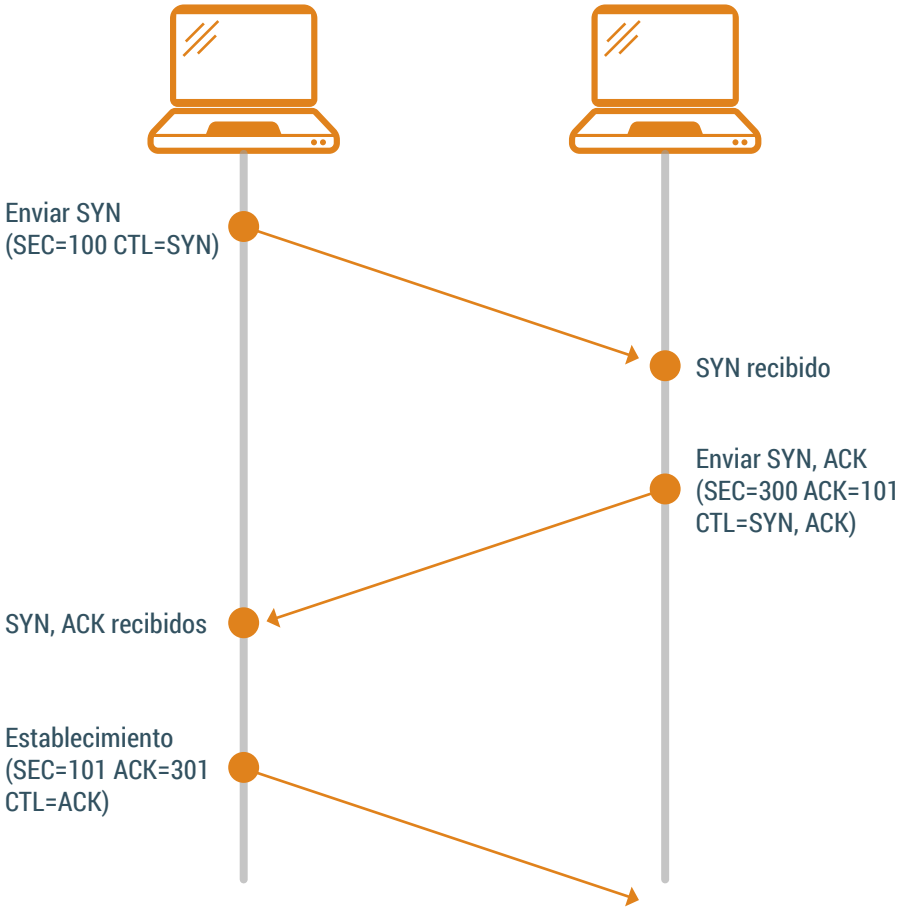


FIGURA 2.3.2.2 CONEXIÓN TCP

Números de secuencia

Cada segmento lleva un número de secuencia, y dicho número de secuencia numera bytes, no segmentos. Es decir, identifica el número de orden del primer byte del segmento. Cuando la conexión se establece se elige el número de secuencia inicial para que no haya confusión entre segmentos que se encuentren en tránsito simultáneamente, aunque sean de comunicaciones diferentes.

Números de asentimiento

La máquina receptora tiene que informar de la correcta recepción de segmentos, pero NO hay necesidad de enviar un asentimiento cada vez que recibimos un segmento. Tenemos la posibilidad de esperar a la recepción de varios segmentos y enviar posteriormente el asentimiento.

El número de asentimiento informa del número de secuencia del próximo byte que es esperado por parte del receptor y tiene un campo específico en cada segmento, así cuando envíe información podrá aprovechar el segmento para añadir el asentimiento en el mismo.

Un aspecto importante es que cada extremo de la comunicación usa sus números de secuencia (partiendo de la misma secuencia inicial) y confirma la recepción (asiente) de los que el otro extremo está usando.

Ventana deslizante

Ventana deslizante es el nombre del protocolo de ventana utilizado para la coordinación en el envío de segmentos y asentimientos. La máquina receptora informa del tamaño de ventana en el campo correspondiente del segmento, está información viene dada en número de bytes, que son los que el receptor tiene la capacidad de recibir. El emisor entonces transmitirá en base a ese número de bytes. A medida que la máquina receptora recibe información tiene la posibilidad de enviar el asentimiento o de deslizar la ventana con el objetivo de la recepción de más bytes. Dado que el envío de datos es bidireccional habrá una ventana en cada extremo de la comunicación.

En el siguiente ejemplo, la máquina emisora ha enviado hasta el byte número 9 pero todavía no ha recibido ningún asentimiento del receptor.

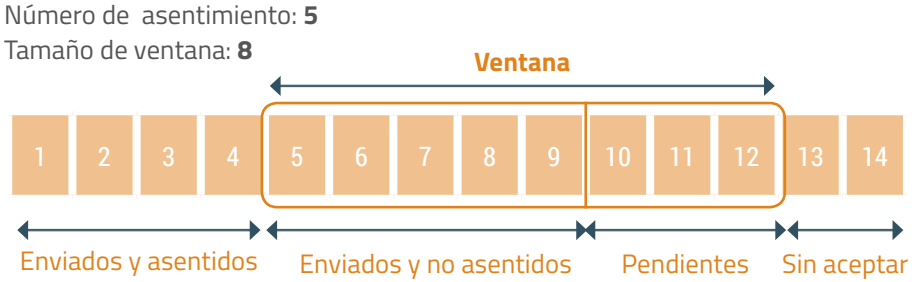


FIGURA 2.3.2.3 VENTANA DESLIZANTE

El tamaño de ventana es variable y permite también ejercer la función de control de flujo de extremo a extremo.

Tiempos de asentimiento

¿Cuánto tiempo es el máximo para recibir un asentimiento entonces?

Para establecer el plazo de asentimiento se usa un algoritmo adaptativo de forma que optimiza la transmisión. En cada envío de segmento se hace el cálculo del RTT (Rount Trip Time) que es igual a la diferencia de tiempo entre la llegada del asentimiento y el envío del segmento. Suele establecerse un tiempo máximo de dos veces el RTT medio.

CONSIDERACIONES

- Plazo para asentimientos: 5 tics.
- El segmento con el byte 2202 se reenvía cuando pasan 5 tics desde que se envió por primera vez.
- El ACK con que responde B engloba todos los datos que ya tiene B.
- A no reenvía el segmento con el byte 2402 pues le llega un ACK que lo engloba antes de que pase el plazo de los 5 tics desde que se envió.
- Los dos últimos ACK que tiene que enviar B pueden agruparse en un único segmento.

Ejemplo completo de conexión

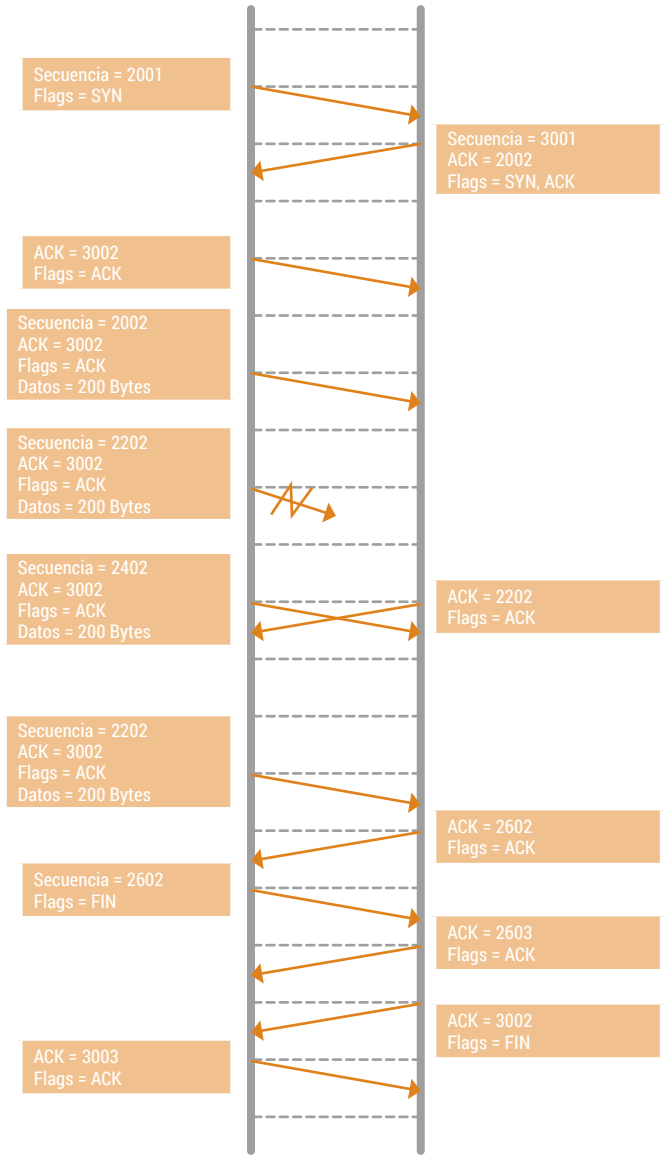


FIGURA 2.3.2.4 EJEMPLO CONEXIÓN TCP

5. NAT

NAT (Network Address Translation) es la solución planteada al agotamiento de direcciones IP, y consiste en dar sólo una IP pública a cada router NAT responsable de la conexión de una sub-red con Internet. Dicho router es responsable de asignar las direcciones IP privadas a cada dispositivo que se conecte a él, debiendo ser diferentes dentro de esa sub-red pero siendo indiferente si es igual a una IP perteneciente a otra red diferente.

Básicamente es un proceso de reescritura por parte de un router de campos de la cabecera IP de los datagramas que encamina:

- Convierte dirección IP y puerto origen en el tráfico que sale.
- Convierte dirección y puerto destino en el tráfico que sale.

El NAT es el responsable de que una determinada organización pueda usar direcciones privadas y tener, en cambio, una IP global de cara a Internet.

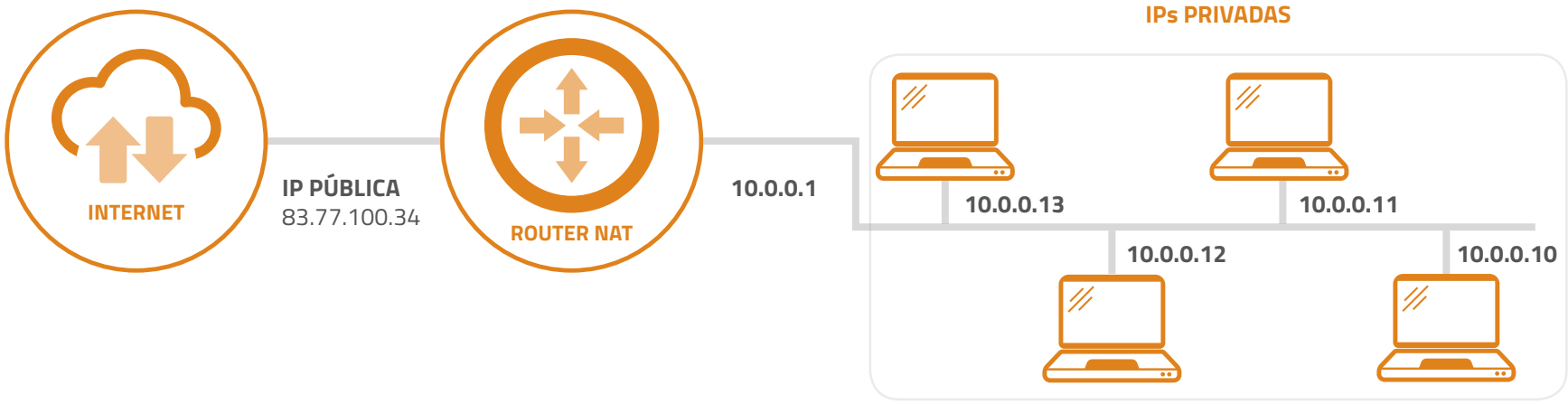


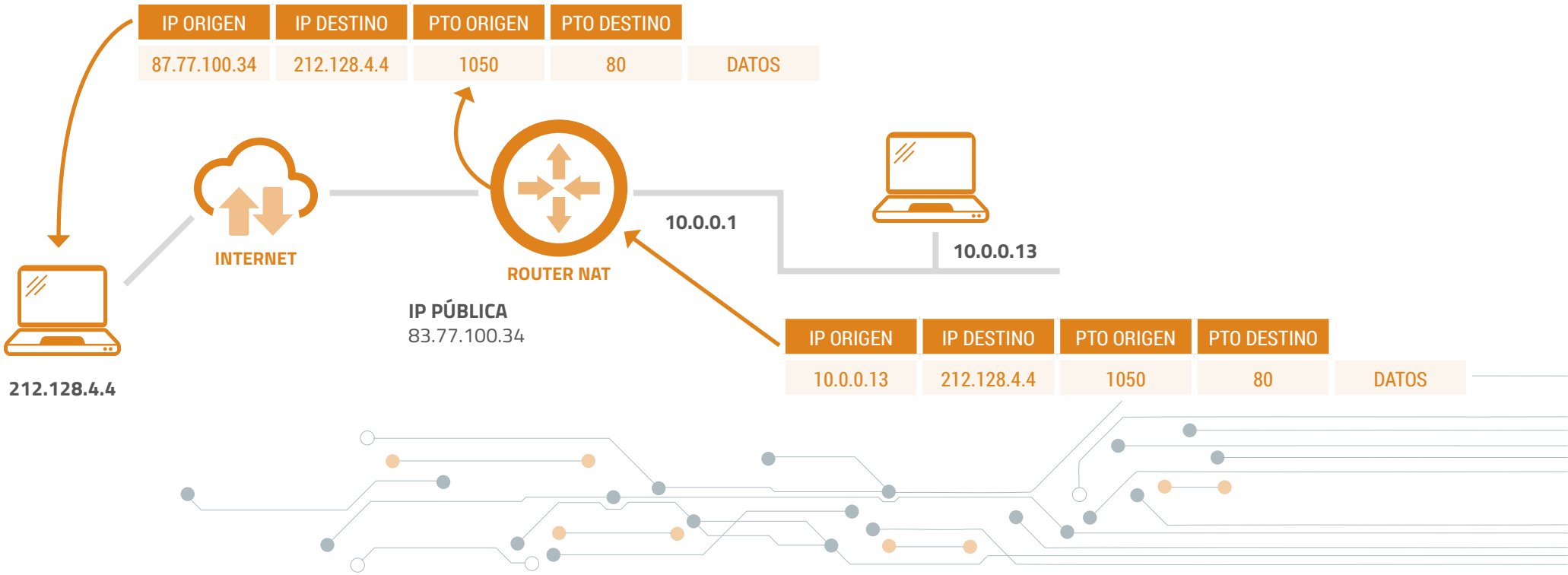
FIGURA 2.4.1 ESQUEMA NAT

En el esquema anterior podemos ver la estructura tipo de NAT, en el que todos los ordenadores de la subred de ese router NAT usan Direcciones IP privadas, que no tienen validez en Internet. El router NAT tiene una dirección válida en Internet (podría tener varias, utilizamos NAT para todos los casos en los que hay menos Direcciones IP públicas que Direcciones IP internas).

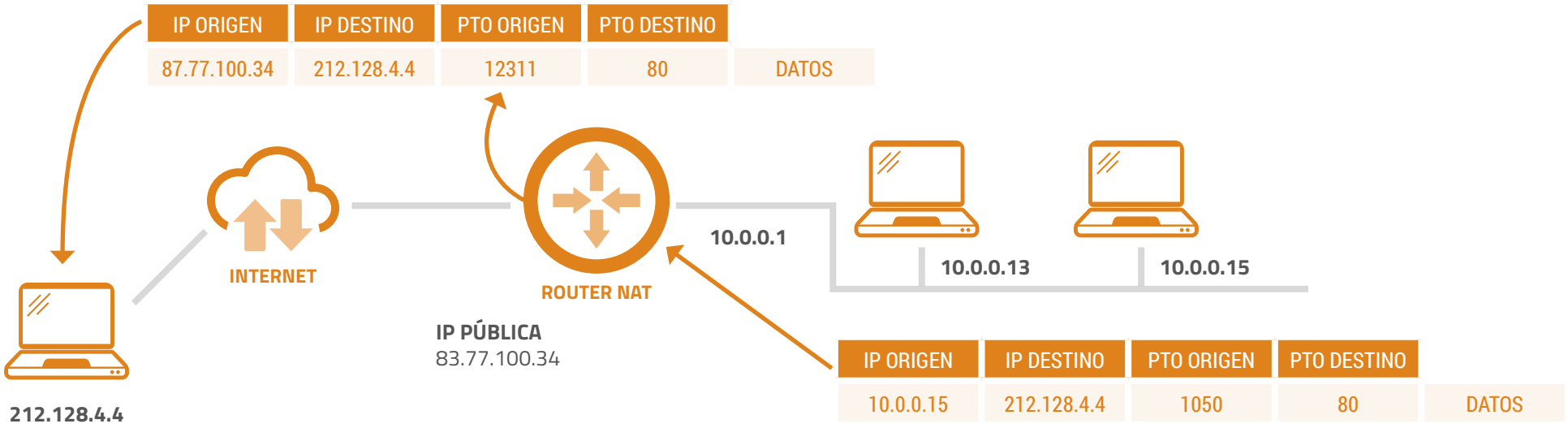
Tráfico saliente

El router NAT sustituye la IP de origen cambiando la IP correspondiente de la subred privada que creó el paquete, por la IP pública que el router tiene asignada.

La máquina que recibe el datagrama piensa que el origen del paquete es el router NAT.



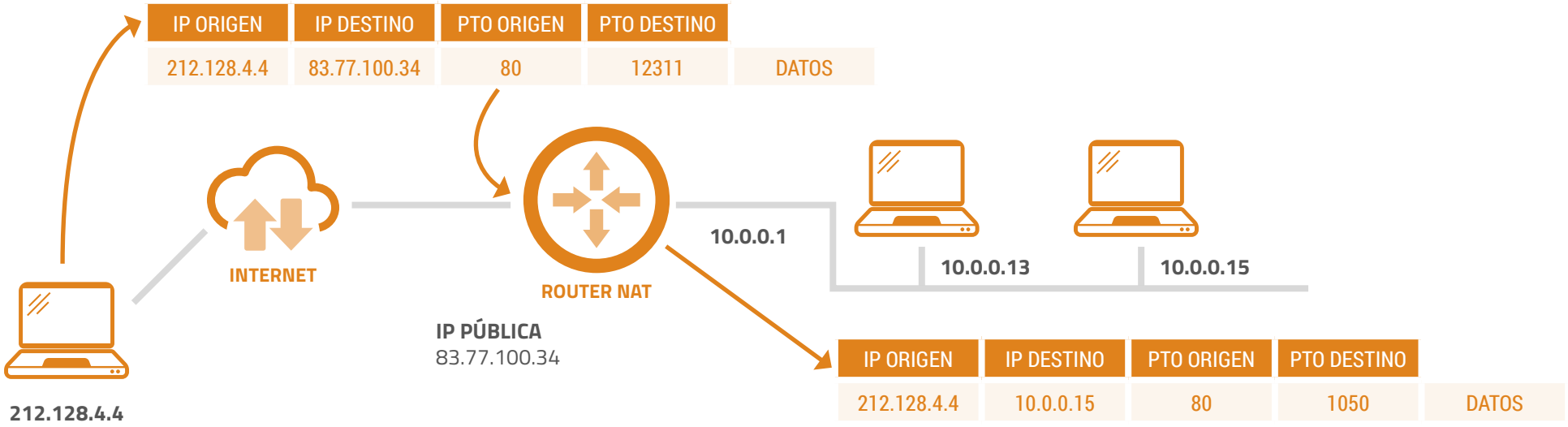
El router NAT tiene una tabla con las sustituciones, cambios y relaciones que se llevan a cabo debido al tráfico saliente, para que en futuras ocasiones se puedan hacer los debidos cambios cuando haya paquetes entrantes. Si otra máquina de la sub-red privada usa el mismo puerto local de origen, el router NAT escogerá un puerto externo distinto al interno, y sustituirá el puerto origen del datagrama por el nuevo puerto externo escogido. Sustituirá además de la IP de origen, el puerto de origen del paquete. A continuación se muestra el esquema y la tabla NAT del router.



RED PRIVADA		INTERNET		PROT DATOS
IP INTERNA	PTO. INTERNO	IP EXTERNA	PTO. EXTERNO	
10.0.0.13	1050	83.77.100.34	1050	TCP
10.0.0.15	1050	83.77.100.34	12311	TCP

Tráfico entrante como respuesta al saliente

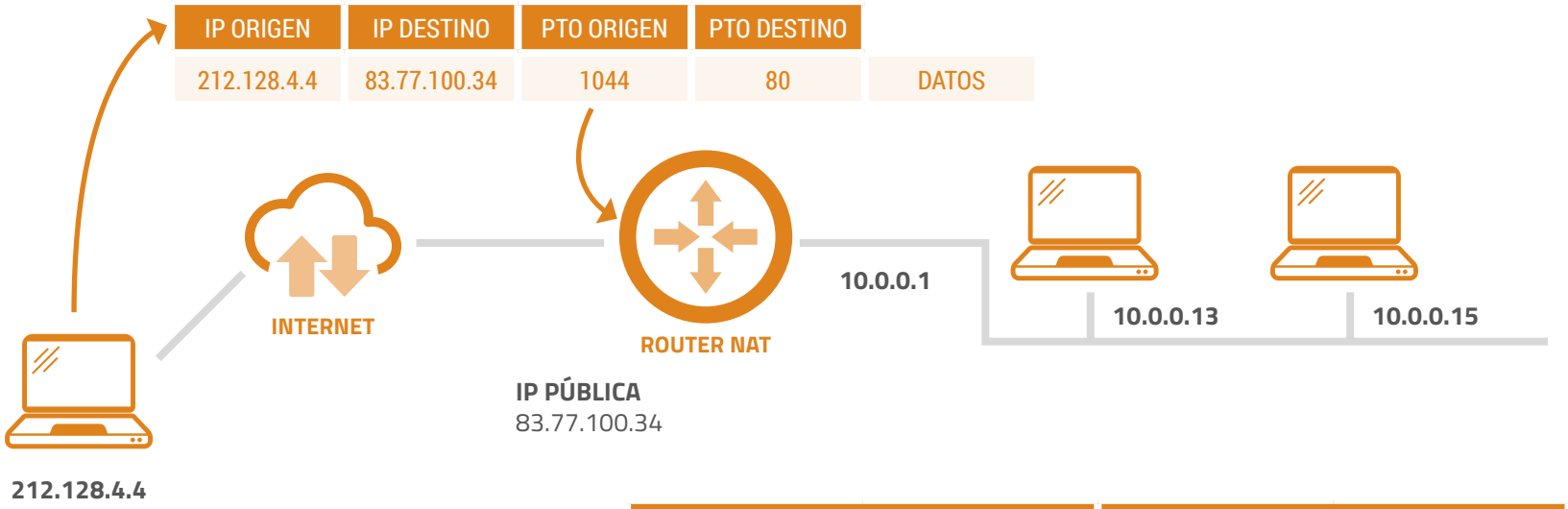
El router NAT sustituye la IP y puerto destino por la IP y puerto internos de los campos de la tabla, cuyos campos IP y Puerto externos sean los de la IP y puerto de destino que trae el datagrama entrante en la sub-red.



RED PRIVADA		INTERNET		PROT DATOS
IP INTERNA	PTO. INTERNO	IP EXTERNA	PTO. EXTERNO	
10.0.0.13	1050	83.77.100.34	1050	TCP
10.0.0.15	1050	83.77.100.34	12311	TCP

Tráfico entrante nuevo

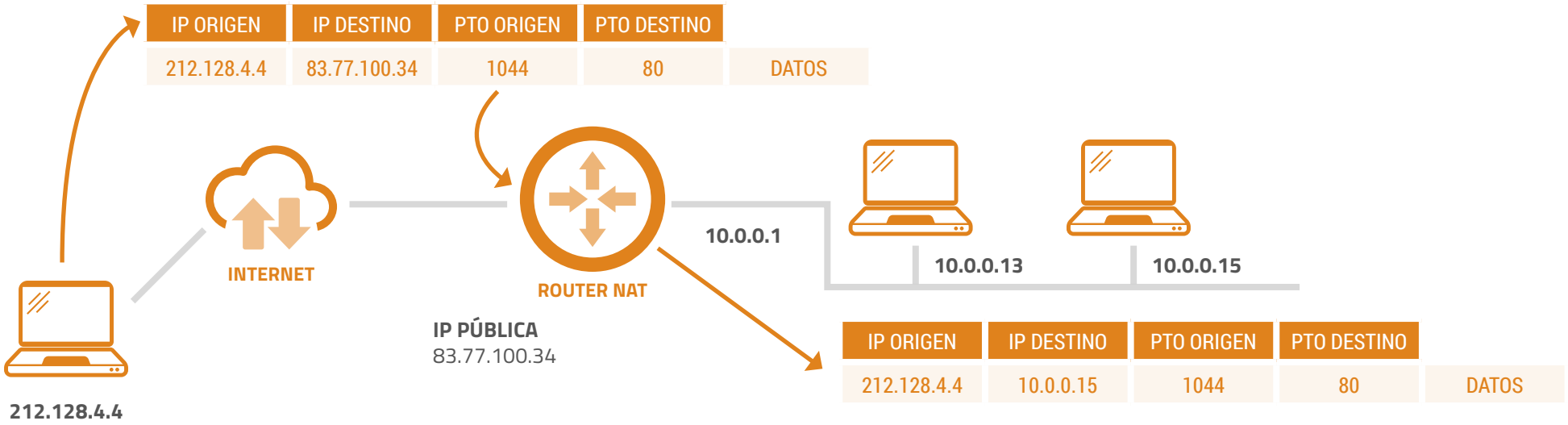
Cuando el router NAT recibe un datagrama nuevo que no responde a un datagrama de salida anterior no tiene información previa para saber a qué máquina interna tiene que redirigirlo así que, por defecto, lo que hace es descartar el datagrama. Es evidente que esto supone un problema, ya que supone que a esta sub-red no le llega nueva información, así que la primera solución consistiría en configurar en el router una IP interna a la que redirigir todos los datagramas cuando no encuentre ninguna entrada que coincida en la tabla.



RED PRIVADA		INTERNET		PROT DATOS
IP INTERNA	PTO. INTERNO	IP EXTERNA	PTO. EXTERNO	
10.0.0.13	1050	83.77.100.34	1050	TCP
10.0.0.15	1050	83.77.100.34	12311	TCP

¡NO HAY ENTRADA CON PUERTO EXTERNO 80!

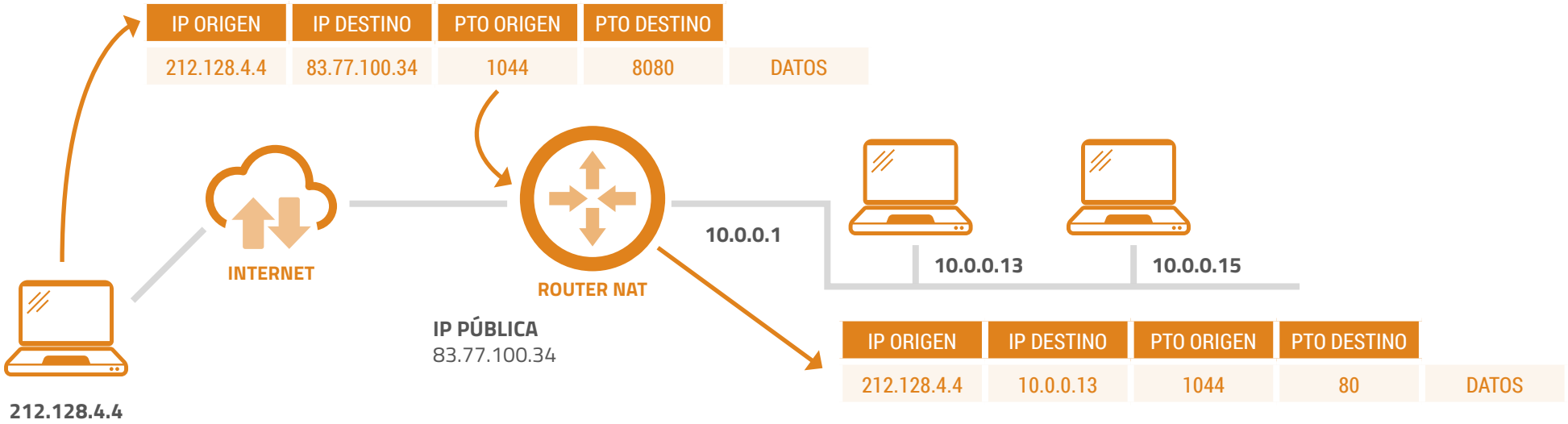
Hay otra solución adicional, que consiste en añadir manualmente (y antes de recibir los datagramas) entradas para los nuevos datagramas con dependencias de la máquina que sea la destinataria. Esta última solución se conoce como "abrir puertos".



RED PRIVADA		INTERNET		PROT DATOS
IP INTERNA	PTO. INTERNO	IP EXTERNA	PTO. EXTERNO	
10.0.0.13	1050	83.77.100.34	1050	TCP
10.0.0.15	1050	83.77.100.34	12311	TCP
10.0.0.15	80	83.77.100.34	80	TCP

Se añade una entrada en la tabla NAT del router, abriendo un puerto. Puede también especificarse un puerto de destino diferente en la sub-red.

De esta forma, se pueden poner en dicha sub-red privada dos servidores en diferentes dispositivos distintos con el mismo puerto. No obstante, desde fuera de la sub-red esos puertos tendrán que tener identificación diferente.



RED PRIVADA		INTERNET		PROT DATOS	
IP INTERNA	PTO. INTERNO	IP EXTERNA	PTO. EXTERNO		
10.0.0.13	1050	83.77.100.34	1050	TCP	> Entrada automática
10.0.0.15	1050	83.77.100.34	12311	TCP	> Entrada automática
10.0.0.15	80	83.77.100.34	80	TCP	> Entrada manual
10.0.0.13	80	83.77.100.34	8080	TCP	> Entrada manual

Inconvenientes

Hay algunos inconvenientes de utilizar este sistema, ya que en algunos protocolos las direcciones IP o puerto de origen se almacenan en la parte de datos del datagrama. En estas situaciones, el router cambia los valores de la cabecera pero sin embargo, no sustituye los valores del campo de datos, así que si la máquina destino coge la información de la parte de datos, no podrán saber el origen correcto de los datos. La única contramedida a este problema es que el router sea consciente del protocolo que se está utilizando para así saber qué datos tiene que modificar en el datagrama completo.

6. Nivel de aplicación

La capa de aplicación es la capa que más cercana está del usuario, ejerciendo de interfaz de las aplicaciones y herramientas que usamos para las comunicaciones y la red subyacente en la que los mensajes son transferidos.
Los protocolos de esta capa se usan con el fin de que los datos fluyan entre las diferentes aplicaciones ejecutadas en la máquina origen y la máquina destino. Hay una gran variedad de protocolos en el nivel de aplicación. Los más conocidos están relacionados con el protocolo de transmisión de hipertexto (HTTP), de archivos (FTP), de acceso a mensajes de Internet (IMAP) y de nombres de dominios (DNS).

6.1 | HTTP

HTTP, o protocolo de transferencia de hipertexto, es un protocolo que se usa para transferir ficheros que conforman las páginas Web de la World Wide Web.

El hipertexto es un texto estructurado que utiliza enlaces lógicos (hipervínculos) entre nodos que contienen texto. Como hemos comentado, HTTP es el protocolo para intercambiar o transferir texto.

HTTP funciona como un protocolo de petición-respuesta en el modelo de computación cliente-servidor. Un navegador web, por ejemplo, puede ser el cliente y una aplicación que se ejecuta en una computadora que aloja un sitio web puede ser el servidor. El cliente envía un mensaje de solicitud HTTP al servidor. El servidor que proporciona recursos como archivos HTML y otro contenido, o realiza otras funciones en nombre del cliente, devuelve un mensaje de respuesta al cliente. La respuesta contiene información de estado de finalización sobre la solicitud y también puede contener contenido solicitado en su cuerpo de mensaje.

Un navegador web es un ejemplo de un agente de usuario (UA). Otros tipos de agente de usuario incluyen el software de indexación utilizado por los proveedores de búsqueda (rastreadores web), navegadores de voz, aplicaciones móviles y otro software que accede, consume o muestra contenido web.

pueden facilitar la comunicación para los clientes sin una dirección globalmente enrutable, mediante la retransmisión de mensajes con servidores externos.

HTTP está diseñado para permitir que elementos de red intermedios mejoren o habiliten las comunicaciones entre clientes y servidores. Los sitios web de alto tráfico a menudo se benefician de los servidores de caché web que ofrecen contenido en nombre de los servidores ascendentes para mejorar el tiempo de respuesta. Los exploradores web almacenan en caché los recursos de la Web previamente accedidos y los reutilizan cuando es posible para reducir el tráfico de la red. Los servidores proxy HTTP en los límites de la red privada pueden facilitar la comunicación para los clientes sin una dirección globalmente enrutable, mediante la retransmisión de mensajes con servidores externos.

HTTP es un protocolo de capa de aplicación diseñado dentro del marco de la suite de protocolos de Internet. Su definición supone un protocolo de capa de transporte subyacente y fiable, y el protocolo de control de transmisión (TCP) se utiliza comúnmente.

Sin embargo, HTTP puede ser adaptado para utilizar protocolos no fiables como el protocolo de datagramas de usuario (UDP), por ejemplo, en HTTPU y Simple Service Discovery Protocol (SSDP).

Los recursos HTTP se identifican y localizan en la red mediante localizadores de recursos uniformes (URL) utilizando los esquemas http y https de identificadores de recursos uniformes (URI). URIs e hipervínculos en documentos HTM forman documentos de hipertexto interconectados.

HTTP /1.1 es una revisión del HTTP original (HTTP /1.0). En esta nueva versión se realiza una conexión independiente para el mismo servidor para cada solicitud de recurso.

HTTP /1.1 puede reutilizar una conexión varias veces para descargar imágenes, scripts, hojas de estilo, etc. Después de que la página haya sido entregada. Por tanto, las comunicaciones HTTP /1.1 experimentan menos latencia, ya que el establecimiento de conexiones TCP presenta una considerable sobrecarga.

6.2 | DNS

Es mucho más cómodo manejar y recordar nombres que direcciones IP mediante números y puntos. Las direcciones IP están ligadas a la estructura de la red pero esto no se reflejará en el nombrado de las diferentes máquinas ya que las direcciones IP están ligadas a máquinas concretas y puede ser muy conveniente un nivel de abstracción independiente de ello (una web puede cambiar de IP sin que necesite cambiar de nombre).

Por todo esto es muy útil establecer un sistema de relación entre nombres de máquinas y sus direcciones IP. Este sistema se denomina DNS (Domain Name System).

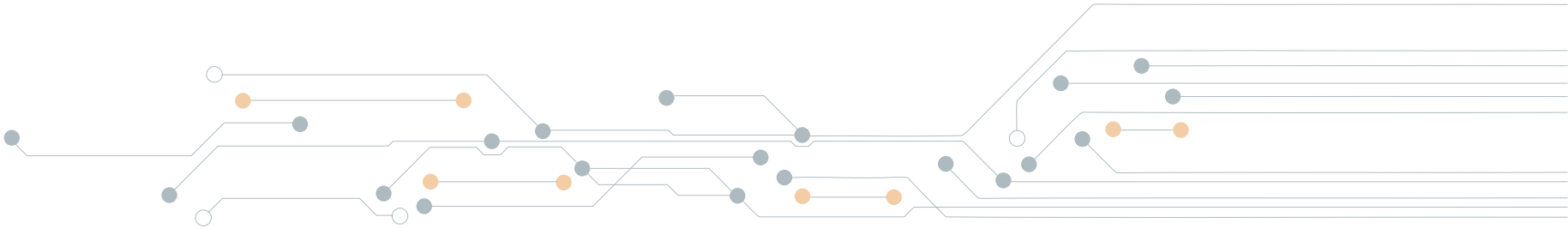
DNS es un protocolo de nivel de aplicación y funciona sobre UDP y TCP. Se trata de una gran base de datos distribuida que es consultada de forma diferente según el modelo cliente/servidor de la red.

Los nombres de las diferentes máquinas se agrupan en dominios, y estos dominios se organizan en forma de árbol.

El nombre del dominio al que pertenece una máquina incluye la unión (separada por puntos) de todos los nombres de dominios. Así, el nombre completo de una máquina está determinado por el nombre de la máquina y el nombre del dominio en el que se encuentra.

Jerarquía de dominios

- Raíz (root domain). Los gestiona ICANN (Internet Corporation for Assigned Names and Numbers) y está servido por root nameservers.
- Primer nivel (TLDS, Top Level Domains). Se trata de los dominios tradicionales (com, org, net...), dominios de la estructura DNS (arpa) y dominios identificativos de cada país (es, uk, de...)
- Segundo nivel
- Tercer nivel



Árbol de dominios

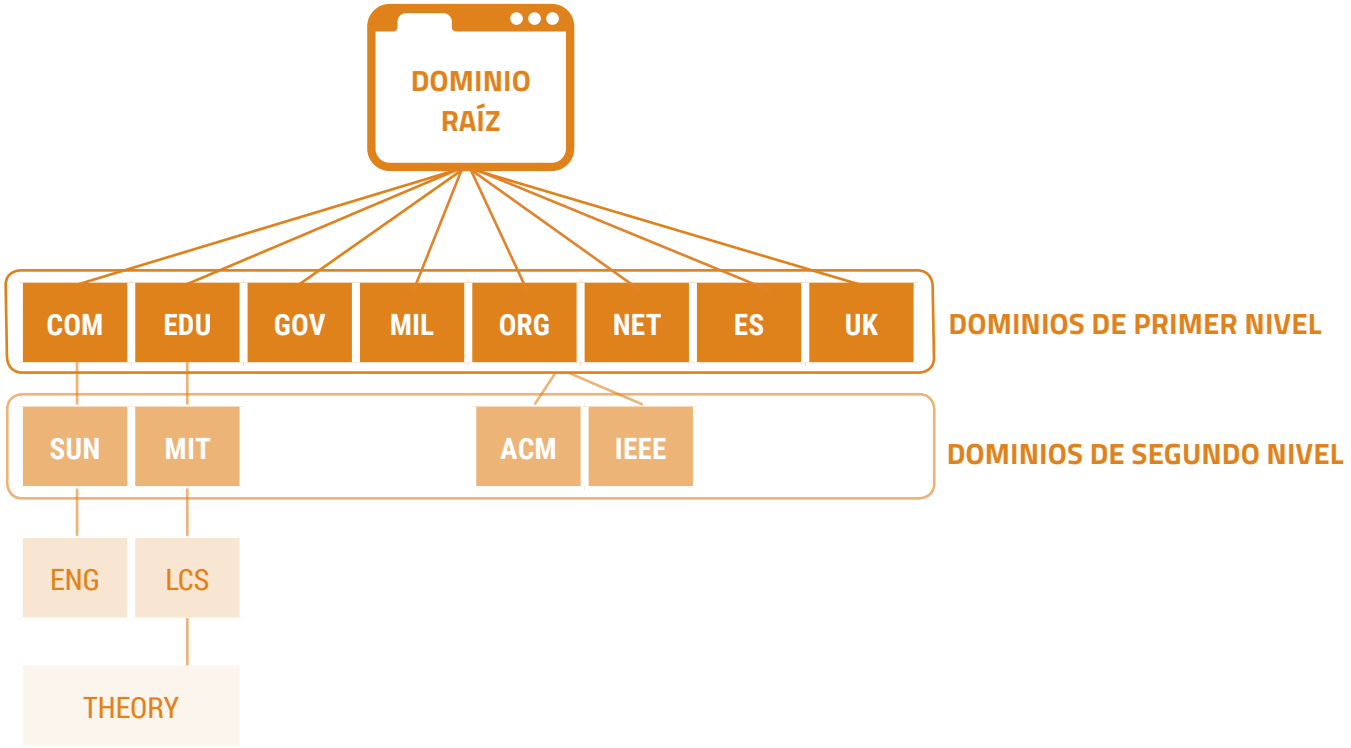


FIGURA 2.5.2.1. ÁRBOL DE DOMINIOS

Los TLDS son asignados por la ICANN, mientras que los nombres de segundo nivel (subdominios .com, .es...) se gestionan a través de "registars" que son dominios registrados. Un subdominio puede estar gestionado por varios "registars".

Resolución de nombres

Cuando una aplicación tiene un nombre de máquina y necesita conocer la IP se realiza una consulta al DNS mediante el comando gethostbyname() o To_IP(), entonces tiene lugar el siguiente proceso:

- Se consulta el fichero /etc/hosts
- Si el nombre no queda resuelto, se realiza una segunda consulta a un servidor DNS (/etc/resolv.conf)
- El fichero /etc/nsswitch.conf decide si es necesario consultar el fichero, el DNS, o ambos.

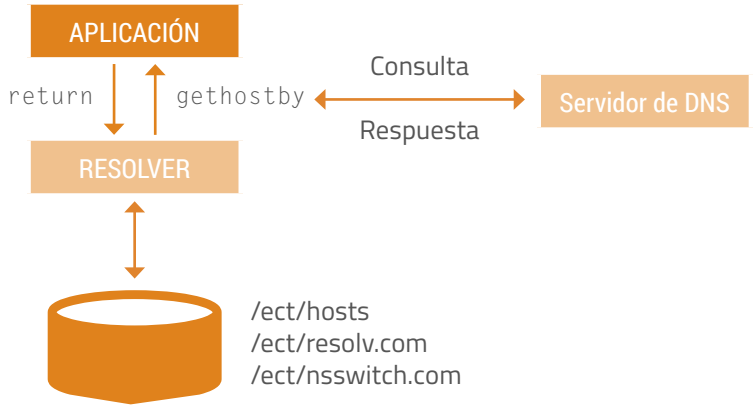


FIGURA 2.5.2.1. RESOLUCIÓN DE NOMBRES

A continuación, se explica el proceso de la consulta de "Resolver" al servidor de DNS.

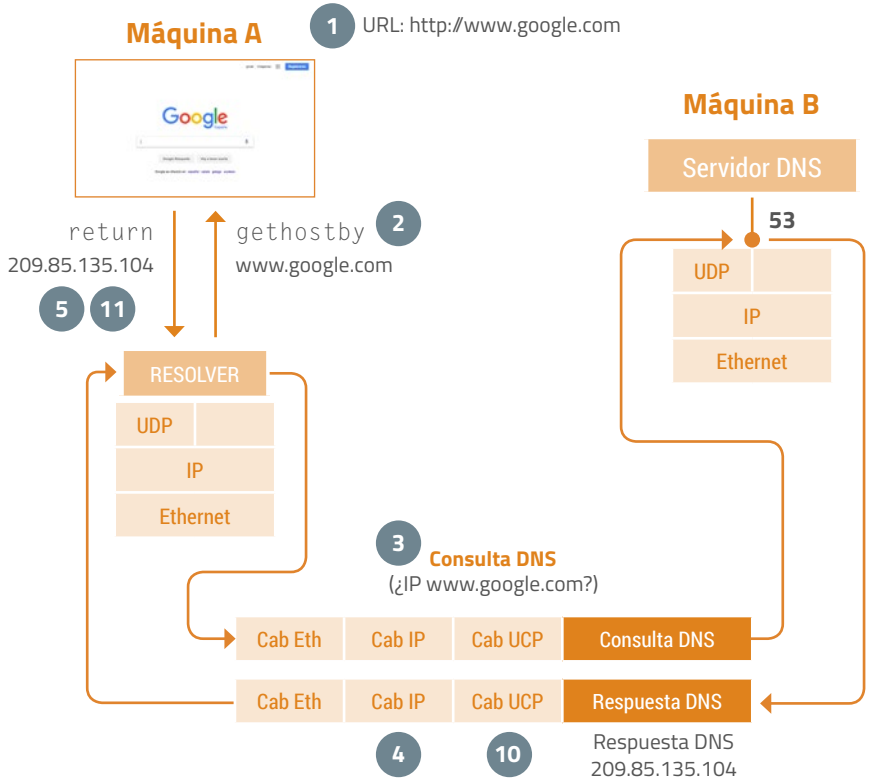


FIGURA 2.5.2.2. RESOLUCIÓN DE NOMBRES

Fuente:http://docplayer.es/18815462-Dns-arquitectura-de-redes-de-ordenadores-arquitectura-de-internet.html

El resolver envía a su servidor DNS con el nombre buscado y el servidor de DNS responde con la IP pedida (si lo sabe contesta directamente, si no, deberá consultar otras fuentes para responder).

La información relacionada con la asignación de diferentes nombres es almacenada en el mapa de dominio, que contiene, como acabamos de comentar, los nombres de máquina con su IP relacionada, así como los subdominios directos y las IP de los servidores DNS de esos

subdominios. Un servidor de DNS que contenga varios ficheros de mapa de dominio sirve a todos esos dominios correspondientes que corresponden a los ficheros. De esta forma, cuando el servidor DNS recibe la consulta, si el servidor de DNS sirve al dominio de la consulta podrá ser consultado directamente, pero si por el contrario, dicho servidor de DNS no sirve al dominio de la consulta, tendrá que responder con la dirección IP de otros servidor de DNS que sirva a un subdominio del dominio de la consulta (esta información sí que puede obtenerla del mapa de dominio).

Por tanto, preguntará a otro servidor DNS con la esperanza de que tenga esa información.

Cuando un servidor recibe una petición (por ejemplo www.google.com) el servidor comprueba si el nombre pertenece a alguno de los dominios que tiene almacenados y devuelve la IP, si por el contrario el nombre no está almacenado:

- El servidor pregunta a otro servidor del dominio raíz que le responderá con la IP de otro servidor del TLD.
- El servidor pregunta al servidor proporcionado, que le contestará con la IP de un servidor de DNS de dominio de segundo nivel.
- Si tuviera más dominios se repite el proceso hasta obtener la IP del dominio en el que está la máquina preguntada.
- El servidor consulta al servidor de último nivel. Este ya sí que podrá proporcionar la IP pedida al servidor.
- El servidor proporciona la IP a la máquina que la solicitó.

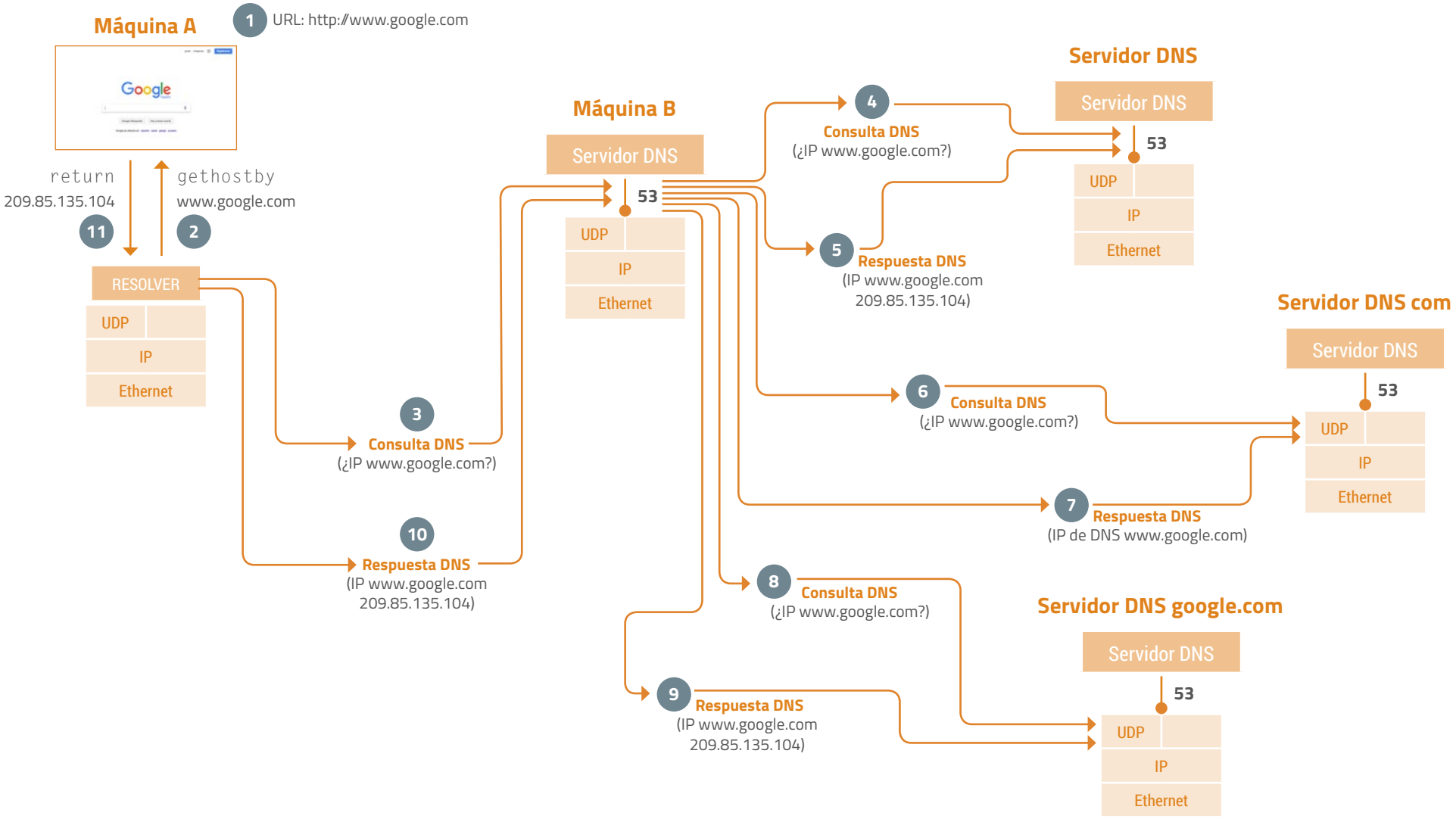


FIGURA 2.5.2.2. RESOLUCIÓN DE NOMBRES

Fuente:http://docplayer.es/18815462-Dns-arquitectura-de-redes-de-ordenadores-arquitectura-de-internet.html

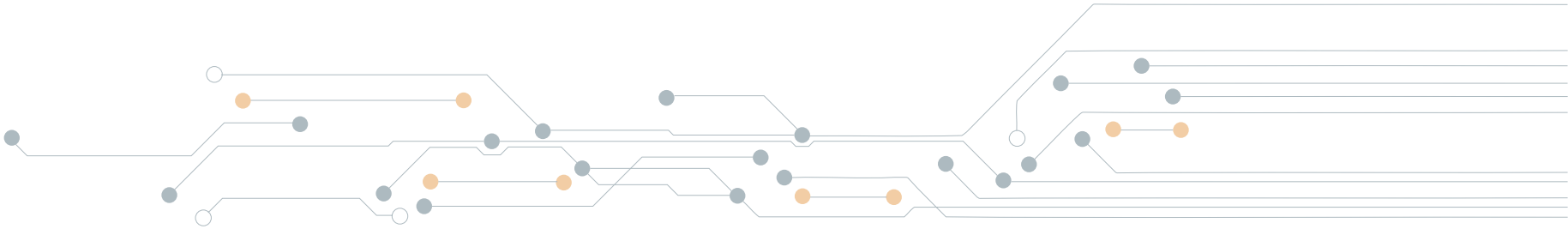
6.3 | SSH

Secure Shell (SSH) es un protocolo criptográfico de red para operar los servicios de red de forma segura a través de una red no segura. El ejemplo de aplicación más conocido es el acceso remoto a sistemas informáticos por parte de los usuarios.

SSH proporciona un canal seguro a través de una red no segura en una arquitectura cliente-servidor, conectando una aplicación cliente SSH con un servidor SSH. Las aplicaciones comunes incluyen la conexión de línea de comando remota y la ejecución de comandos remotos, pero cualquier servicio de red puede protegerse con SSH. La especificación del protocolo distingue entre dos versiones principales, denominada SSH-1 y SSH-2.

La aplicación más visible del protocolo es el acceso a cuentas Shell en sistemas operativos tipo Unix, pero también ve un uso limitado en Windows. En 2015, Microsoft anunció que incluiría soporte nativo para SSH en una futura versión.

SSH fue diseñado para protocolos de Shell remoto no seguros. Estos protocolos envían información, en particular contraseñas, en texto plano, haciéndolos susceptibles a la interceptación y la divulgación mediante el análisis de paquetes. El cifrado utilizado por SSH está destinado a proporcionar confidencialidad e integridad de los datos a través de una red no segura.



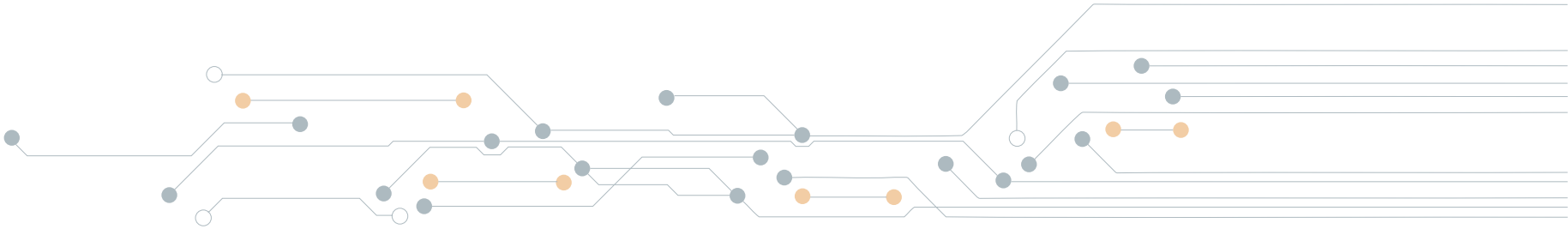
SSH utiliza criptografía de clave pública para autenticar el equipo remoto y permitir que autentique al usuario si es necesario. Hay varias maneras de usar SSH:

- Una es usar pares de claves pública-privada generadas automáticamente para cifrar simplemente una conexión de red, y a continuación utilizar la autenticación de contraseña para iniciar sesión.
- La otra es usar un par de claves pública-privada generadas manualmente para realizar la autenticación, permitiendo a los usuarios o programas iniciar sesión sin tener que especificar una contraseña. En este escenario cualquiera puede producir un par de claves diferentes (Públicas y privadas). La clave pública se coloca en todos los equipos que deben permitir el acceso al propietario de la clave privada coincidente. Aunque la autenticación se basa en la clave privada, la clave en sí nunca se transfiera a través de la red durante la autenticación. SSH sólo verifica si la misma persona que ofrece la clave pública también posee la clave privada coincidente. En todas las versiones de SSH es importante verificar claves públicas desconocidas, es decir, asociar las claves públicas con identidades, antes de aceptar como válidas. Aceptar la clave pública de un atacante sin validación autorizará a un atacante no autorizado como usuario válido.

Características de SSH

El protocolo de SSH tiene entonces las siguientes protecciones:

- Tras la conexión inicial, el cliente es capaz de verificar si su conexión es en el mismo servidor de otra conexión anterior.
- El cliente envía datos de autenticación al servidor utilizando encriptación robusta de 128 bits.
- La información mandada y recibida en la sesión es transferida con encriptación de 128 bits (difícil de descifrar).

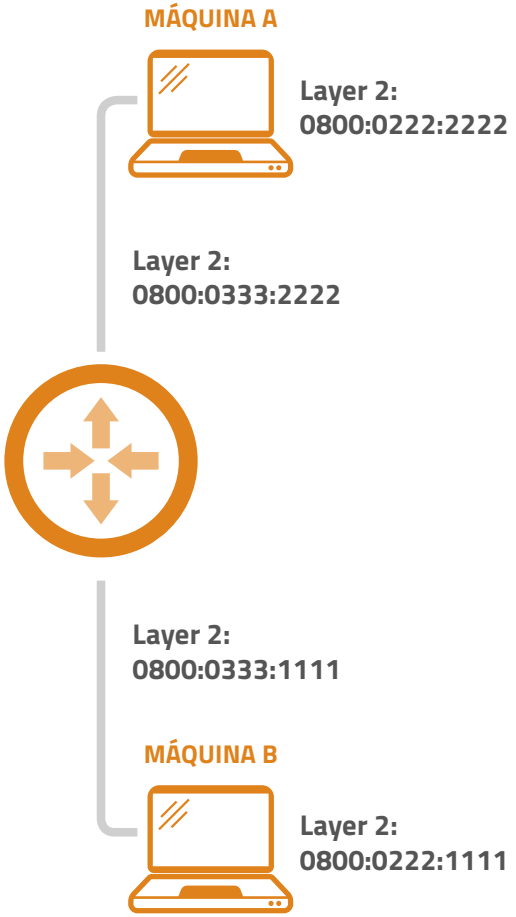


7. Ejemplo final

Todos los conceptos aprendidos anteriormente pueden ponerse en práctica estudiando el proceso de enrutamiento de paquetes, observando la correlación entre capa 2 y capa 3.

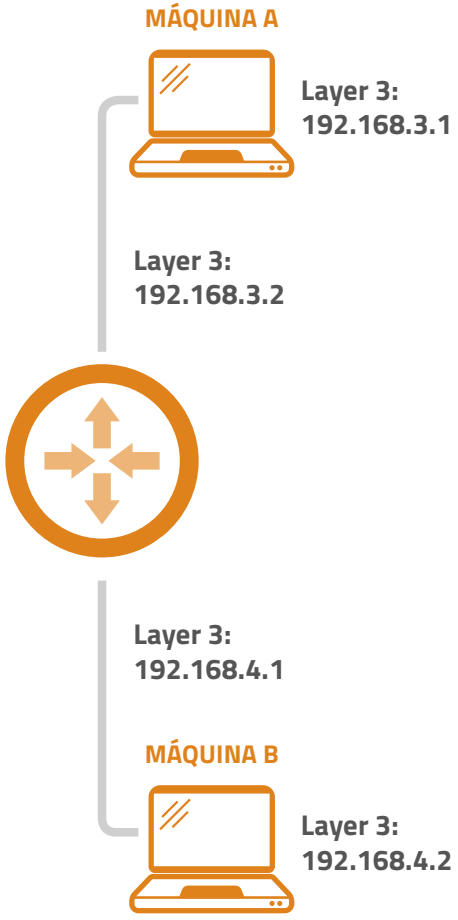
Direccionamiento en capa 2. Direcciones MAC

Al explorar el proceso de entrega de paquetes vamos a tener en cuenta este esquema, con un router entre ambas máquinas. Nuestro ejemplo va a tener las direcciones MAC de los dispositivos mostrados en la figura. Recuerde que tendremos que resolver las direcciones MAC del router para el envío de paquetes entre máquinas de diferentes segmentos.



Direccionamiento en capa 3. Direcciones IP

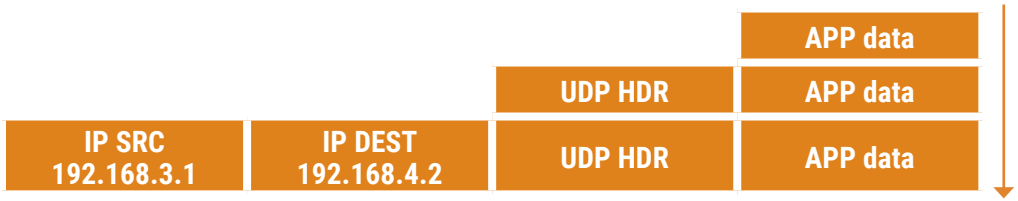
En esta figura se observa la vista de capa 3, incluyendo las direcciones IP de los diferentes dispositivos. Este diseño puede deberse al deseo de dividir la red en dos segmentos por razones de seguridad, rendimiento. Se va a estudiar sólo la función de enrutamiento, sin tener en cuenta posibles funciones de seguridad, cortafuegos o filtrado.

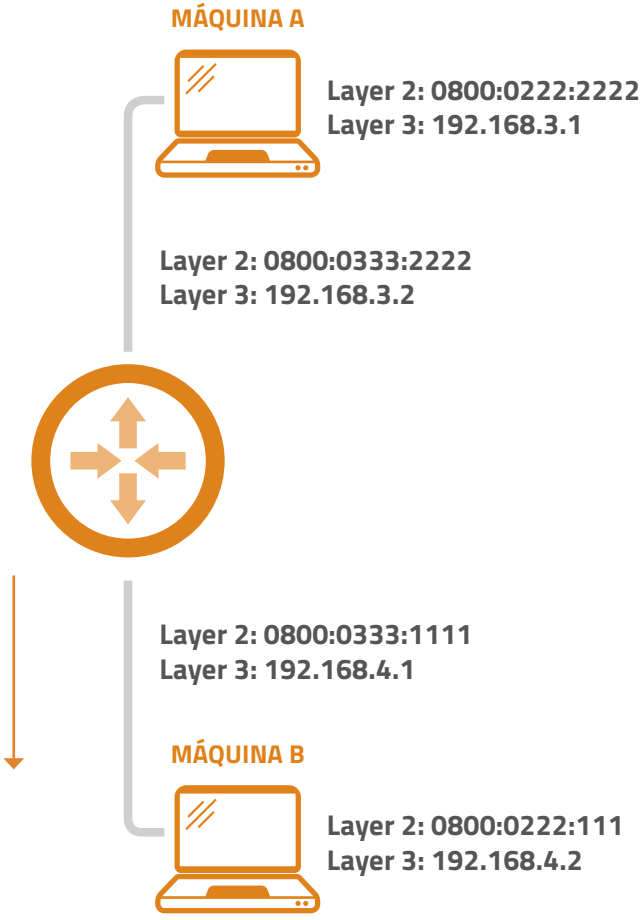


IP ROUTING. Proceso de envío de paquetes

El primer paso que se debe realizar es la resolución de nombres DNS (en el caso de que usen DNS), convirtiendo dicho nombre a una dirección IP y escogiendo el protocolo de transporte que se va a usar. En el ejemplo explicado se va a utilizar UDP. Cada capa agregará su propia etiqueta en forma de encabezado.

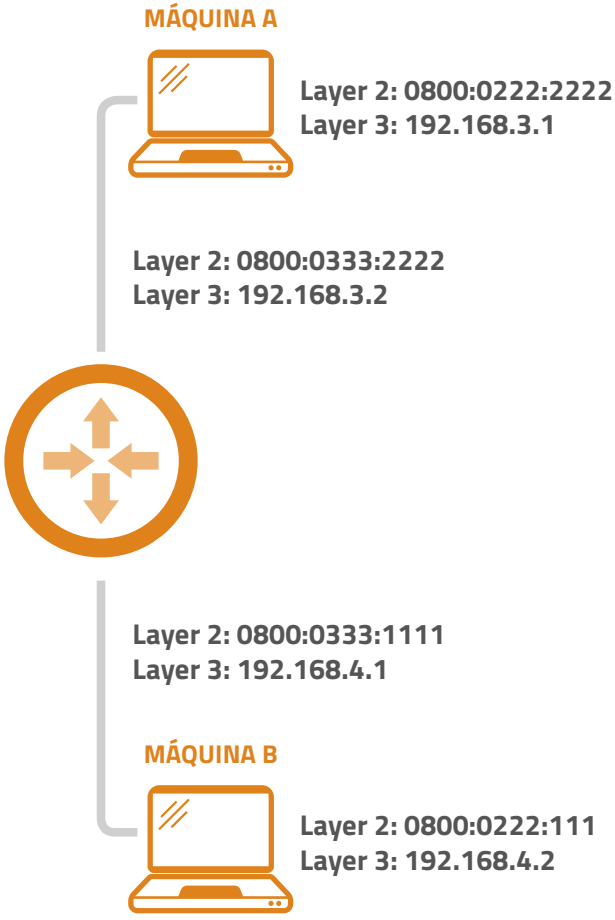
De esta forma, la capa de aplicación identifica que tiene que mandar un paquete a la dirección 192.168.4.2 y que no necesita una conexión fiable. La capa de transporte por tanto decide establecer una conexión UDP y necesita que la capa 3 añada la información necesaria para enviar el paquete. En la capa 3 se pondrá el encabezado IP y luego se pedirá a la capa 2 que envíe el paquete.





La capa 2 responde alegando que no tiene información sobre esa dirección IP, ya que no tiene la dirección MAC y por tanto va a necesitar resolver esa información mediante una solicitud ARP. El paquete permanecerá bloqueado sin enviarse hasta que no se complete el proceso de solicitud de ARP. En este punto del proceso, justo entre la capa 2 y la capa 3, el dispositivo dirá "Según esta dirección IP y está máscara, el destino está en una red diferente, ya que yo estoy en la red 192.168.3 y el destino está en la red 192.168.4".

Los tres primeros bytes de la dirección IP identifican la red, así que el proceso ARP dice algo parecido a "No necesito resolverlos para dirección MAC del destino. No soy un router y no sé cómo enviar esto, pero mi puerta de enlace predeterminada lo sabrá, así que voy a intentar resolver la dirección MAC de la puerta de enlace predeterminada, que se encuentra en la configuración del protocolo IP del dispositivo".

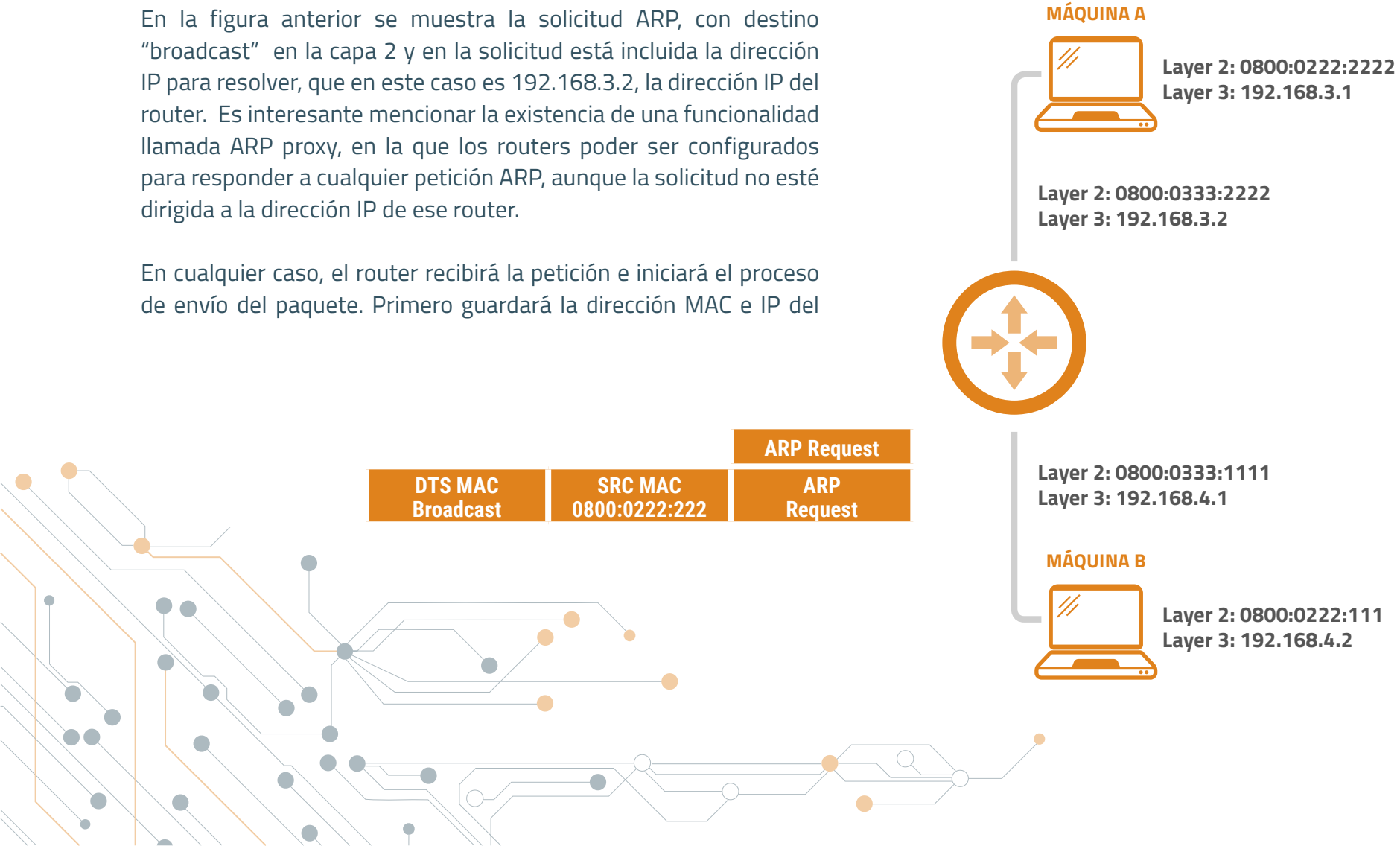


Esta es probablemente una de las primeras y más comunes fuentes de errores y en la solución de problemas es necesario asegurarnos de que la dirección IP de la puerta de enlace predeterminada correcta esté configurada. Si no existe conocimiento de dónde enviarlo o qué router debe procesar esto, el paquete no llegará allí.

En la figura anterior se muestra la solicitud ARP, con destino "broadcast" en la capa 2 y en la solicitud está incluida la dirección IP para resolver, que en este caso es 192.168.3.2, la dirección IP del router. Es interesante mencionar la existencia de una funcionalidad llamada ARP proxy, en la que los routers poder ser configurados para responder a cualquier petición ARP, aunque la solicitud no esté dirigida a la dirección IP de ese router.

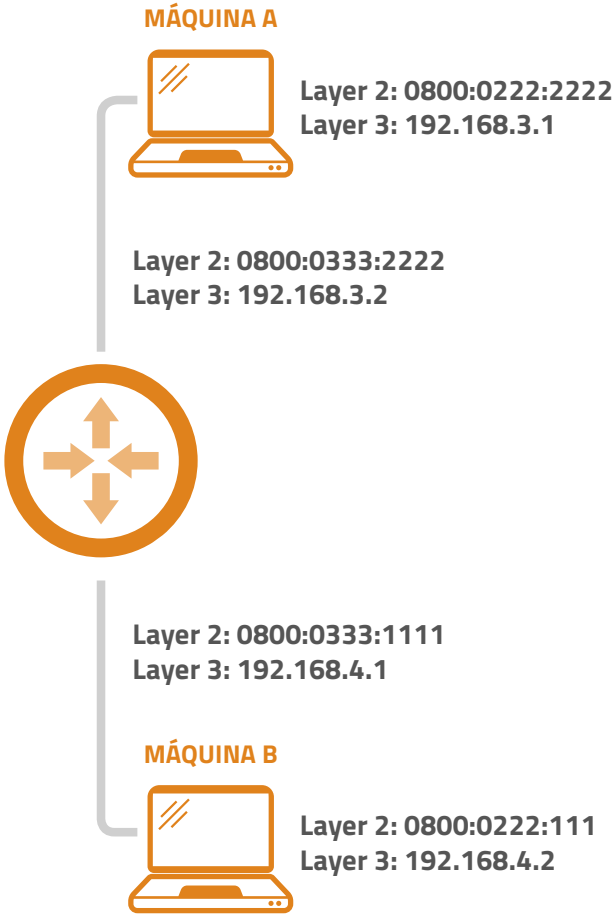
En cualquier caso, el router recibirá la petición e iniciará el proceso de envío del paquete. Primero guardará la dirección MAC e IP del emisor en su propia tabla ARP. El router es un dispositivo IP como cualquier otro, por lo tanto cumplirá todas las reglas de IP.

En la siguiente figura se muestra el envío del paquete por parte del router.



Desde este punto, el router enviará una respuesta ARP indicando su dirección MAC y la posibilidad de que se le envíen paquetes a él. Ahora la dirección del emisor está almacenada en su tabla ARP, vinculando la dirección IP de la puerta de enlace a la dirección MAC de la puerta de enlace. Por tanto, está listo para enviar los paquetes hacia su destino.

DTS MAC	SRC MAC	ARP
0800:0222:2222	0800:0333:222	Request



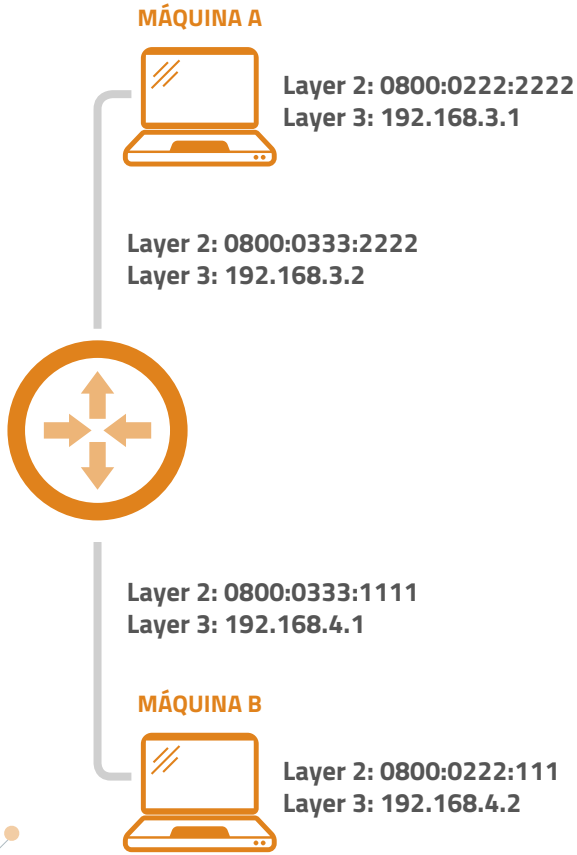
El proceso ARP habrá obtenido un ARP reply de 192.168.3.2, añadiendo esa IP y MAC a su tabla ARP, tendrá una asignación para su puerta predeterminada (192.168.4.2) y tendrá asignada la IP 192.168.4.2 a la MAC 0800:0333:2222.

Recuerde que esas entradas eventualmente se agotarán y que el proceso ARP puede repetirse a lo largo de la conversación dependiendo de los tiempos de inactividad y los tiempos absolutos.

El paquete que estaba en espera se libera y se envía utilizando la dirección IP del destinatario, la dirección IP de origen del remitente, el MAC de origen del remitente y el MAC de destino es la dirección MAC del router. El router sólo funciona en la capa 3, por lo que verá el paquete entrando. Lo digiere y lo procesa, porque está destinado a sí mismo en términos de dirección MAC en la capa 2.

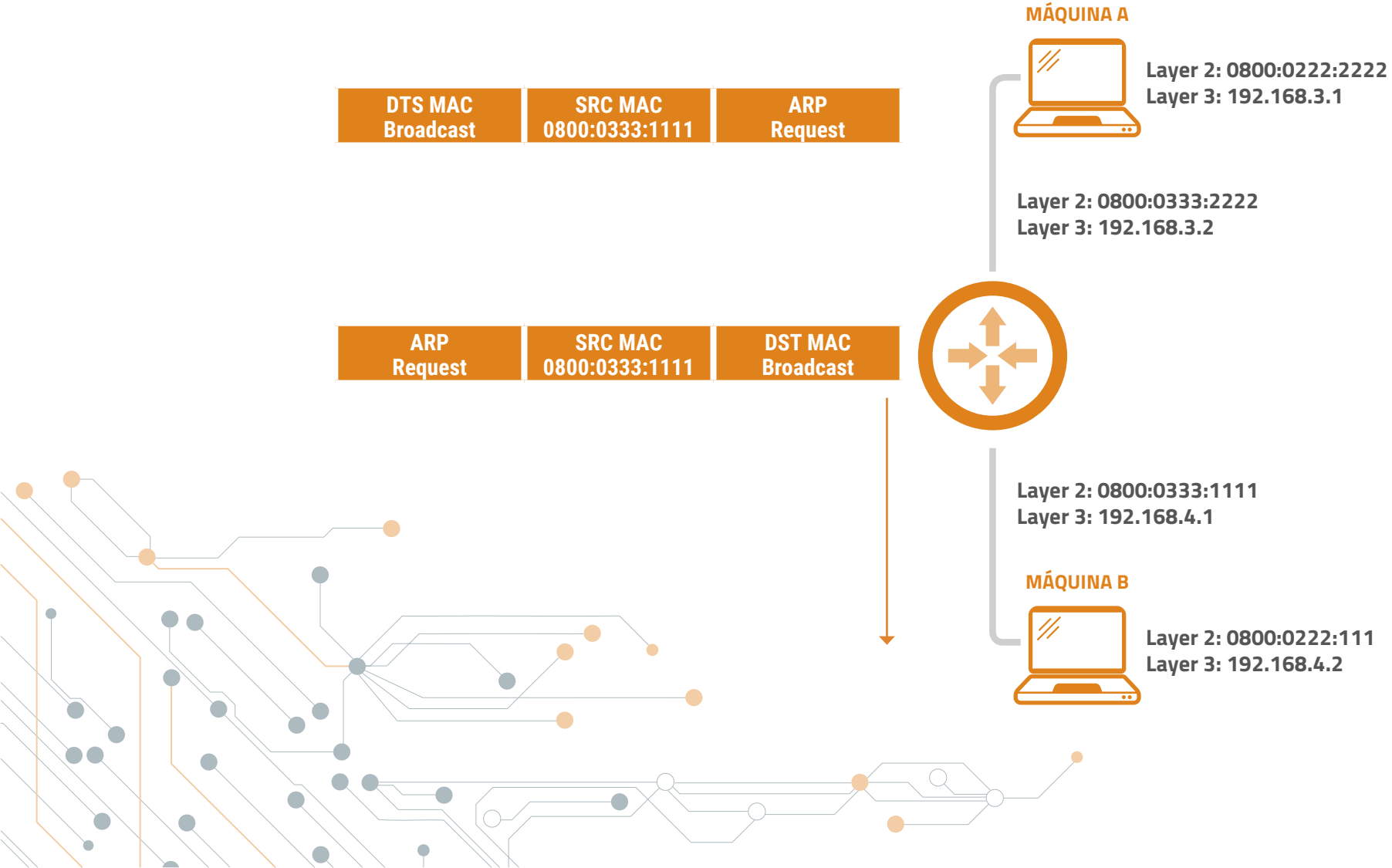
Se desencapsulará y enviará a la capa 3, donde tiene lugar la función de encaminamiento y reenvío. Es por eso que aunque la dirección IP de destino no es la del router, el router reenviará de acuerdo a su tabla de enrutamiento.

APP DATA	UDP HD	DTS IP 192.168.4.2	SRC IP 192.168.3.1	SRC MAC 0800:0333:222	DST MAC 0800:0222:2222

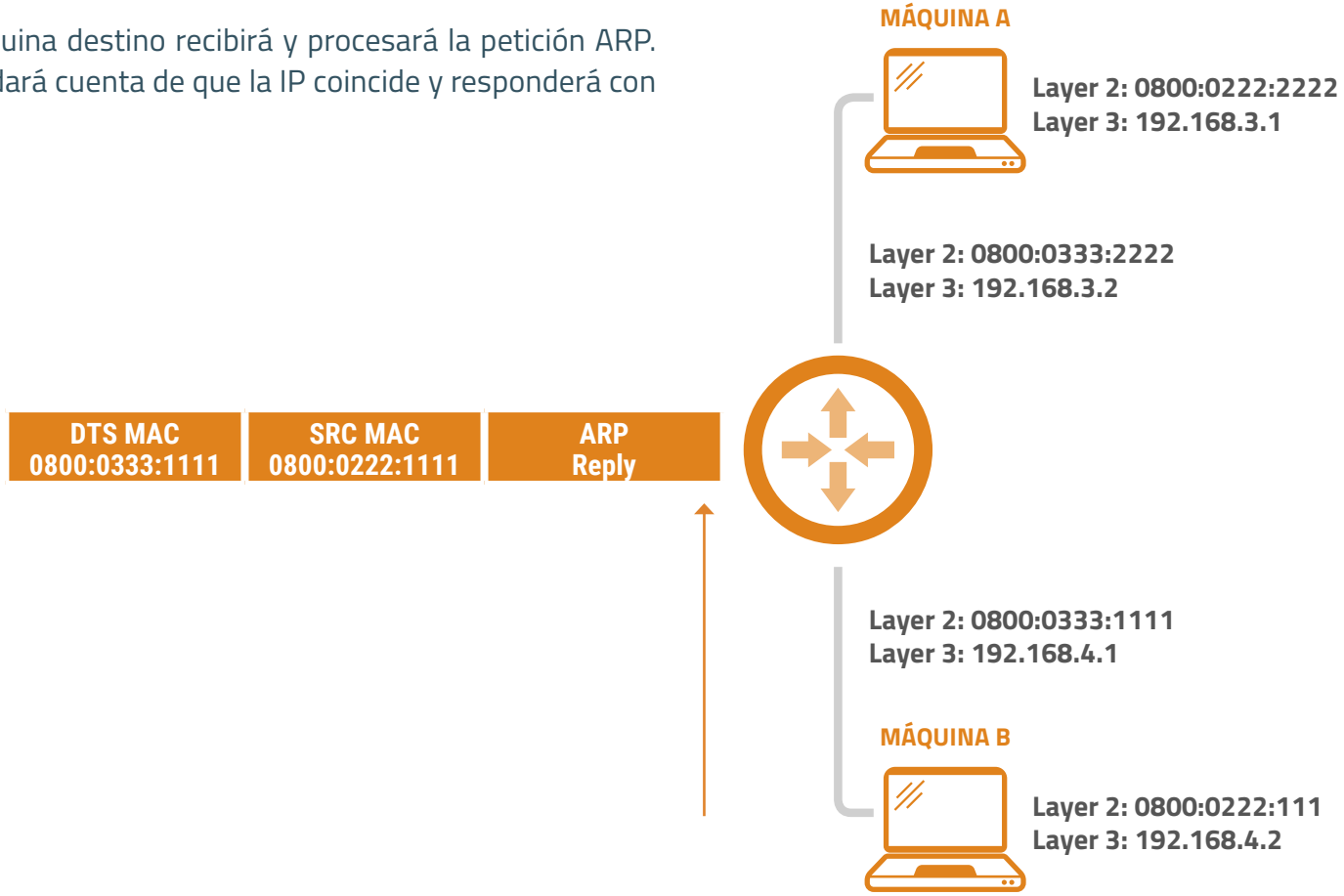


Al explorar la tabla de enrutamiento, el router se dará cuenta de que la IP destino es una entrada de su tabla. 192.168.4.0 con la máscara apropiada, está conectada directamente, así que enviará directamente al proceso de capa 2 y resolver la dirección MAC del destino, mediante otro proceso ARP. Si no se tratara de un destino del mismo segmento, la entrada de la tabla de enrutamiento

En ese momento, el router pediría el reenvío a ese dispositivo intermedio y así la resolución ARP iría contra ese dispositivo para encontrar su propia dirección MAC. En este caso el escenario es más siempre con dos redes conectadas.

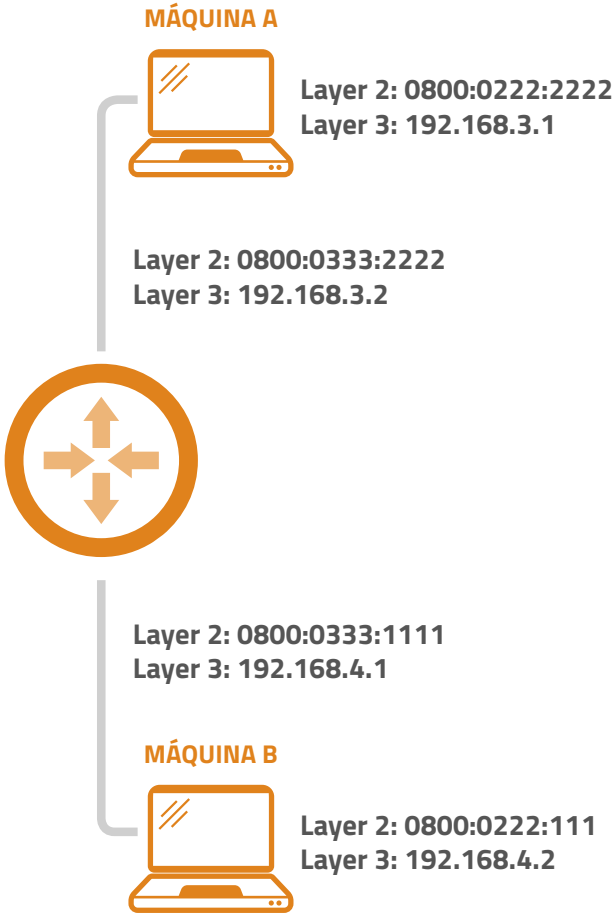


Por tanto, la máquina destino recibirá y procesará la petición ARP. Rápidamente se dará cuenta de que la IP coincide y responderá con su dirección MAC.



Antes de que se envíe la respuesta ARP, la máquina destino también guardará la asignación de la IP del router a su MAC en la tabla ARP. Es interesante ver cómo las máquinas llenarán la tabla ARP, no sólo cuando vean una respuesta ARP, sino cuando vean una solicitud ARP también.

El router verá la respuesta ARP, sabrá la dirección MAC de la máquina destino y estará listo para montar el paquete completo con las direcciones IP de origen y destino así como las MAC de origen y destino.



Por lo tanto, las comunicaciones IP en redes remotas no son más que el trabajo incremental de una serie de intermediarios llamados routers que reenviarán el tráfico de acuerdo a cierta inteligencia. Sin embargo, el proceso global en término de ARP, asignaciones, etc., es exactamente el mismo.

Telefónica EDUCACIÓN DIGITAL