Instalación de un Remote Rig sin dirección IP pública

L.M. Esteban, EA2BFM; H. Sánchez, EA2DR y F. Tusell, EA2BEE

Introducción

En un trabajo previo del tercer autor se describió la instalación de una estación remota para modos digitales. Examinamos entonces las diferentes opciones de conectividad, que a veces se limitan a un acceso a internet vía *router* 4G. Vimos que la utilización de una red privada virtual (VPN) permite obviar el problema de que la dirección IP del módem 4G es típicamente inaccesible a conexiones iniciadas desde el exterior, lo que dificulta conexiones entrantes. De hecho, el manual del Remote Rig RRC-1258 MkII especifica en su página 241 que es requisito indispensable para su utilización disponer de una dirección IP pública en el lado de la radio.

Esto en ocasiones no es posible y, cuando es posible, puede ser relativamente caro. En lo que sigue examinamos qué puede hacerse en estos casos y documentamos un modo de hacerlo.

Se presupone alguna familiaridad con el sistema operativo Linux, pero solo conocimientos básicos sobre redes. Obviamos todo lo referente a la configuración de un RemoteRig, suficientemente explicado en el manual de usuario, concentrándonos en el aspecto concreto del acceso cuando se carece de una IP pública. Para no hacer este trabajo excesivamente largo, detalles inesenciales se desplazan a la página web https://ea2bee.ure.es/RR/CompRR.html.

La raíz del problema

Esta sección puede ser obviada por lectores familiarizados con redes bajo protocolos TCP/IP y el modo de operar de NAT (Network Address Translation).

Dentro de nuestra red doméstica, típicamente coexistirán varios dispositivos, todos manteniendo conexiones (incluso simultáneas) con Internet. Nuestro *router* canaliza todas estas conexiones a través de una línea de cobre, fibra, o inalámbrica. Si nuestro *router* tiene asignada la dirección 88.14.59.215, por ejemplo, desde el exterior todas las conexiones se ven como emanando de esa dirección. El *router*, sin embargo, "lleva la cuenta" de qué tráfico procede de (o tiene por destino a) cada una de las máquinas en nuestra red local doméstica y canaliza las respuestas a la máquina apropiada.

Para ello reescribe las direcciones de origen y destino de los paquetes salientes y entrantes, en el proceso conocido como NAT (Network Address Translation).

Sin embargo, si nuestro *router* recibe un paquete del exterior *que no es respuesta a una conexión iniciada por una de nuestras máquinas*, no tiene modo de saber a quién dirigirla. Puede hacerlo si hemos configurado los puertos especificando algo como: «cualquier paquete al puerto 22, dirígelo a la máquina 192.168.1.35». O si situamos una máquina en la «zona desmilitarizada» (DMZ), lo que la convierte en receptora de cualquier paquete para el que nuestro *router* no cuente con instrucciones concretas.

El CG-NAT (Carrier Grade NAT) es otra capa que efectúa un proceso conceptualmente similar al descrito. Nuestro *router* doméstico comparte su dirección 88.14.59.215 quizá con otros muchos, que se conectan a un "macro-*router*" encaminando todo el tráfico desde y hasta las redes locales que sirve. Se entiende pues que si una máquina externa pretendiera iniciar una conexión a 88.14.59.215, ese "macro-*router*" no sabría a quién redirigirla. La dirección 88.14.59.215 es por ello no pública.

Algunos, no todos, los proveedores de internet asignan direcciones públicas a sus *routers* cableados; prácticamente nadie lo hace a las tarjetas telefónicas, que dan lugar a conexiones de datos con IP no pública. Por ello, dispositivos detrás de un *router* 4G servido por una tarjeta telefónica no serán directamente accesibles desde el exterior, aunque puedan iniciar conexiones al exterior. Este es el problema que se plantea al situar un RemoteRig tras un *router* 4G.

Un caso práctico

En URV, sección en Bizkaia de URE, teníamos el proyecto de situar en una de las ubicaciones de repetidores que mantenemos un transceptor "todo modo", con mayor funcionalidad y prestaciones que el QDX a que se refiere un artículo anterior. Tanto en la ubicación en el monte ("monte" en lo sucesivo) como en nuestra sede social ("sede") carecemos de acceso cableado a Internet, por lo que se hacía preciso recurrir a *router*s 4G en ambos extremos.

Podría parecer que el establecimiento de una red privada virtual (VPN) da solución al problema, del modo que se describió en el artículo precedente. No es así: hay aquí un importante detalle adicional. Los equipos que se trata de enlazar (las unidades "Radio" y "Control" de un sistema Remote Rig RRC-1258 MkII) no admiten la instalación de *software* de VPN sobre ellos.

El problema al que nos enfrentamos aquí es el de enlazar, no dos equipos dentro de una VPN, sino dos equipos que no pueden formar parte de la VPN.

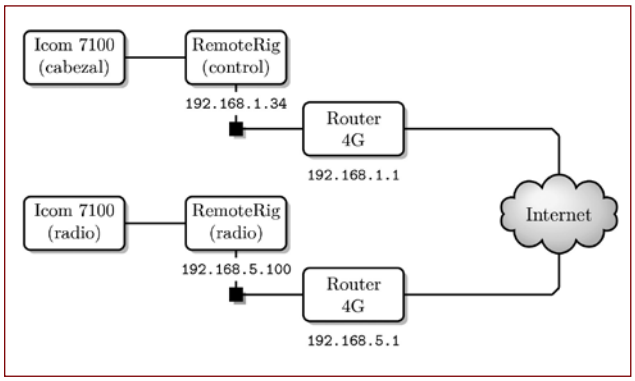

Figura 1. El problema: hacer que *se vean* las dos mitades de un Remote Rig (RR Control, y RR Radio) situadas en sendos segmentos de red detrás de *routers* 4G

Las dos mitades del RemoteRig (figura 1) cuelgan de segmentos de red que tienen salida a Internet mediante *routers* 4G con dirección no pública. Desde ninguno de ellos se puede iniciar una conexión al otro porque las direcciones IP con que se sale a Internet no son publicas al estar detrás de un CG-NAT.

Si pudiéramos instalar *software* de red privada virtual (VPN) en las dos unidades del RemoteRig, el problema podría solventarse fácilmente; pero no podemos porque no son piezas de hardware en que el usuario pueda instalar tal *software*. La idea que inmediatamente acude, sin embargo, es simple: instalar un ordenador en cada segmento que actúe como puente, enlazando ambos ordenadores en una VPN.

La configuración resultante una vez que dichos dos microordenadores se integran en los respectivos segmentos puede verse en la figura 2. La disposición de los equipos y sus conexiones pueden verse en las figuras 3 y 4.

Cualesquiera ordenadores capaces de correr *software* de VPN y enrutar paquetes sirven. Por todos los motivos nos interesará utilizar sistema operativo Linux. Por economía, simplicidad y estabilidad conviene utilizar máquinas tipo Raspberry Pi, de muy bajo consumo (en torno a 5 W), sin partes móviles como discos duros, ventiladores, etc. y por tanto con una durabilidad muy grande. Pueden ser iguales o diferentes: en nuestro caso son una Raspberry 2 en el segmento local ("sede") y una Orange Pi 2E en el segmento remoto ("monte"), pero es simplemente porque es lo que teníamos a mano.

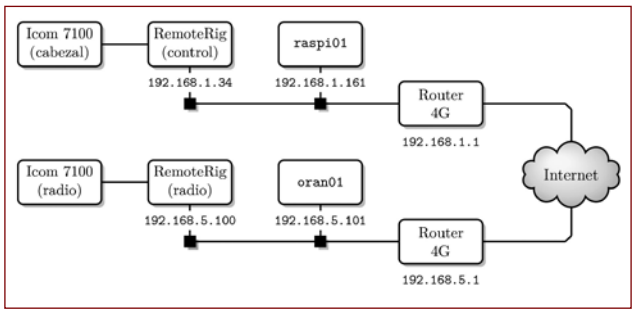

Figura 2. raspi01 y oran01 son micro-ordenadores que permiten enlazar los dos segmentos vía una VPN. En nuestro caso, el primero es una Raspberry 2 y el segundo una Orange Pi 2E, pero pueden ser cualquier cosa que admita *software* de VPN y enrutamiento

Figura 3. De izquierda a derecha, cuerpo del Icom 7100, RemoteRig, router 4G y microordenador Orange Pi (oran01). Son la mitad "monte" de nuestra instalación

■ Vimos que la utilización de una red privada virtual (VPN) permite obviar el problema de que la dirección IP del módem 4G es típicamente inaccesible a conexiones iniciadas desde el exterior. [...] Examinamos qué puede hacerse en estos casos y documentamos un modo de hacerlo

Figura 4. De izquierda a derecha, Raspberry Pi (raspi01), router 4G, RemoteRig y cabezal del Icom 7100. Son la mitad "sede" de nuestra instalación

Una vez conectados entre sí los elementos en los dos segmentos de red procederemos como sigue.

Paso 1: establecimiento de direcciones IP estables

Para lo que sigue (en particular, para el establecimiento de rutas) necesitamos que cada elemento tenga una dirección IP estable. Lo más simple es permitir que el *router* las asigne mediante DHCP. Para ello, en cada uno de los *routers* daremos de alta los respectivos equipos, emparejando su dirección MAC con la dirección IP deseada. La Figura 5 muestra la pantalla de configuración en uno de los *routers* utilizados (TP-Link Archer MR6400).

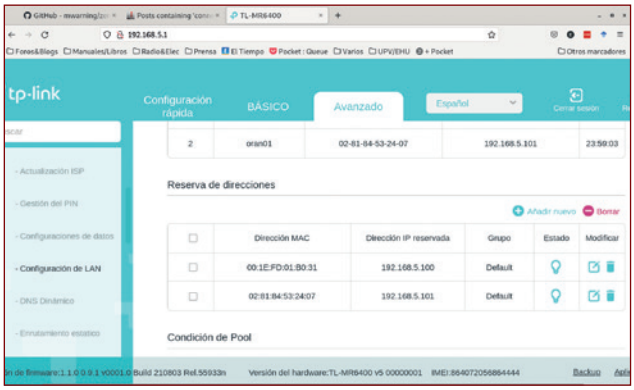

Figura 5. Asignación de direcciones en el "monte". La dirección 192.168.5.100 corresponde al RemoteRig, la 192.168.5.101 a oran01, el microordenador que sirve de enrutador

Los dos segmentos de red no deben compartir prefijo; es decir, no emplearemos direcciones 192.168.1.x en ambos segmentos; si en uno tenemos direcciones 192.168.1.z, en el otro fijaremos 192.168.y.z (en nuestro ejemplo, y=1 en el segmento "sede" e y=5 en el segmento "monte").

Paso 2: creación de una red privada virtual (VPN)

Para que `oran01` y `raspi01` puedan verse mutuamente los integraremos en una VPN (ver figura 2). Recordemos que ambos equipos no son directamente accesibles si los *routers* no tienen dirección pública; pero ambos pueden establecer conexiones salientes a un servidor de VPN, que los pondrá mutuamente en relación.

Esto lo podemos hacer con casi cualquier tipo de VPN, pero se impone una máquina en el exterior con IP pública a la que ambas máquinas puedan conectar. Si empleamos SoftEther (ver enlaces en los complementos *online* a este artículo) hemos de proveer esta máquina. Hay otras VPN que nos dan todo integrado y permiten configurar la VPN en minutos. Nosotros hemos empleado ZeroTier. Es un servicio comercial, pero de fuente abierta y gratuito para redes de hasta 25 nodos (ampliamente suficiente, para nuestro uso). Si accedemos a su página web podemos, tras darnos de alta, crear una VPN de forma visual, en unos pocos clics sobre una página web.

Comenzará por pedirnos un nombre para la nueva VPN y nos atribuirá un Network ID o identificador, como `af78b–f9436e1880c`. A continuación, nos ofrecerá un surtido de direcciones para que escojamos el rango de las que ocupará nuestra VPN (en el ejemplo, figura 6, hemos escogido el segmento 10.147.17.*x*).

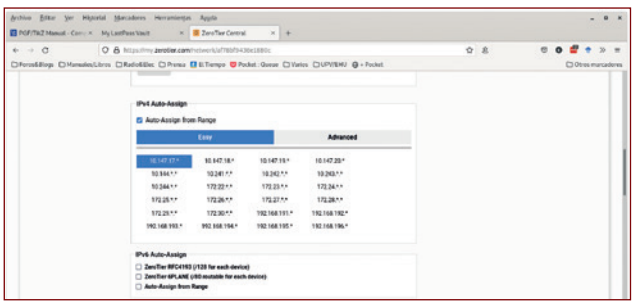

Figura 6. Selección de direcciones para la nueva VPN en la página de creación de ZaroTier

Hecho esto, descargaremos el módulo "cliente" en cada una de las máquinas que deseemos integrar: lo hay para todos los sistemas operativos usuales, incluso puede instalarse en teléfonos.

Tras instalar el cliente en `oran01` y `raspi01` teclearemos en una terminal de dichas máquinas:

```
zerotier-cli join af78bf9436e1880c
```

Regresando a la página de creación de la VPN veremos que nos aparecen las nuevas máquinas integradas, pidiéndonos que marquemos una casilla si las autorizamos. La figura 7 muestra un fragmento de dicha página mostrando alguno de los equipos que venimos comentando (`raspi01`, *online*), así como algunos otros, todos ellos autorizados. Vemos algunas máquinas *online*, otras permanecen apagadas. Puede verse también el teléfono de uno de los autores (*offline*): podemos controlar la red desde del teléfono, lo que tiene utilidad en aplicaciones como la nuestra en que podemos querer verificar el estado de máquinas remotas en todo momento y lugar.

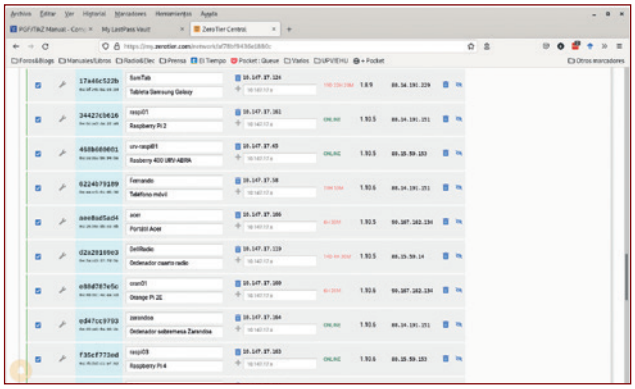

Figura 7. Listado de máquinas en una VPN de ZeroTier. Las direcciones 10.147.17.*x* son las de dentro de la VPN, mientras que en la columna de la derecha tenemos las direcciones reales que pueden o no ser públicas

Integradas todas las máquinas que deseamos formen parte de la VPN, ya podemos acceder desde cualquiera a cualquiera otra (insistimos: aunque una de ellas o las dos estén tras de un CG-NAT). En particular, y retornando a nuestro ejemplo (figura 2), tendremos conectividad desde `oran01` a `raspi01` y viceversa, que podremos comprobar con un ping o simplemente abriendo una sesión sobre una máquina desde la otra. Ello no significa que tengamos solucionado nuestro problema, pues lo que necesitamos es acceso mutuo entre las dos mitades del RemoteRig (192.168.1.34 y 192.168.5.100 en la figura 2). Tenemos todavía trabajo por hacer.

Paso 3: hacer trabajar a oran01 y raspi01 como enrutadores

Podemos examinar los *interfaces* de red de `raspi01` mediante el mandato ifconfig, que requiere privilegios de súper usuario (*root*). En nuestro caso, veremos algo como lo que se reproduce en la figura 8.

`raspi01` es accesible vía VPN en la *interface* zthnhf4hb2. Queremos que todo lo que llega a esa *interface* se vuelque al segmento local, vía *interface* enxb827eb29df7a. Para ello tenemos detalladas instrucciones en la documentación de ZeroTier. Todas las instrucciones que siguen se deben ejecutar como superusuario (*root*).

- Editaremos el fichero `/etc/sysctl.conf` en `raspi01` de manera que acabe conteniendo una línea como: `net.ipv4.ip_forward=1`. (Dicha línea suele estar comentada, por lo que habitualmente bastará retirar el símbolo de comentario # al comienzo.) Para que los cambios surtan efecto es preciso reiniciar la máquina, o bien teclear:

```
sysctl -w net.ipv4.ip_forward=1
```

- Instalaremos, si no lo están ya, los paquetes `iptables` e `iptables-persistent`. En una máquina corriendo Debian, basta teclear:

```
apt-get install iptables iptables-persistent
```

```
root@raspi01:~# ifconfig
enxb827eb29df7a: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.161  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::8aa:baca:8481:6895  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:29:df:7a  txqueuelen 1000  (Ethernet)
        RX packets 8052  bytes 874439 (853.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 9128  bytes 1064400 (1.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 2346  bytes 380539 (371.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2346  bytes 380539 (371.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

zthnhf4hb2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 2800
        inet 10.147.17.161  netmask 255.255.255.0  broadcast 10.147.17.255
        inet6 fe80::cbc:a3ff:fe4a:22a9  prefixlen 64  scopeid 0x20<link>
        ether 0e:bc:a3:4a:22:a9  txqueuelen 1000  (Ethernet)
        RX packets 129  bytes 30434 (29.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 39  bytes 4726 (4.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figura 8. Tarjetas y pseudo-tarjetas de red en la máquina raspi01. Con la dirección 192.168.1.161 aparece el (único) *interface* físico. Con la dirección 10.147.17.161 aparece la *interface* virtual de la VPN

- iptables es una herramienta muy utilizada para instalar cortafuegos o de cualquier modo filtrar el tráfico que recibimos/enviamos desde cada *interface* de red. Tiene una configuración muy flexible mediante reglas. En nuestro caso, lo único que queremos es copiar de una *interface* a otra, por lo que las reglas son muy simples. Teclearemos:

```
PHY_IFACE=enxb827eb29df7a; ZT_IFACE=zthnhf4hb2
iptables -t nat -A POSTROUTING -o $PHY_IFACE -j MASQUERADE
iptables -A FORWARD -i $PHY_IFACE -o $ZT_IFACE -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -i $ZT_IFACE -o $PHY_IFACE -j ACCEPT
```

- Dichas reglas esencialmente copian paquetes de una *interface* a otro, en las dos direcciones. Puede convenirnos crear un pequeño fichero de comandos y ejecutarlo, en previsión de que un error nos obligue a reteclear todo. Para realizar los cambios persistentes teclearemos:

```
bash -c iptables-save > /etc/iptables/rules.v4
```

Realizaremos exactamente las mismas operaciones en la otra maquina encargada de enrutar paquetes en el segmento 192.168.5.x (oran01). Nótese que los nombres de los *interfaces* pueden cambiar de una máquina a otra.

Paso 4: establecer rutas entre los dos segmentos

Completado el paso anterior, los paquetes que lleguen a raspi01 y oran01 vía la VPN serán volcados a sus respectivos segmentos físicos; pero *hemos de hacer que lleguen a esas dos máquinas*, para que puedan hacer su trabajo.

Esto es más simple. No es preciso manipular cada máquina. En la pantalla de control de ZeroTier podemos añadir rutas que se implementan en todas las máquinas de la VPN. La figura 9 muestra un fragmento de dicha página con las rutas que precisamos ya definidas.

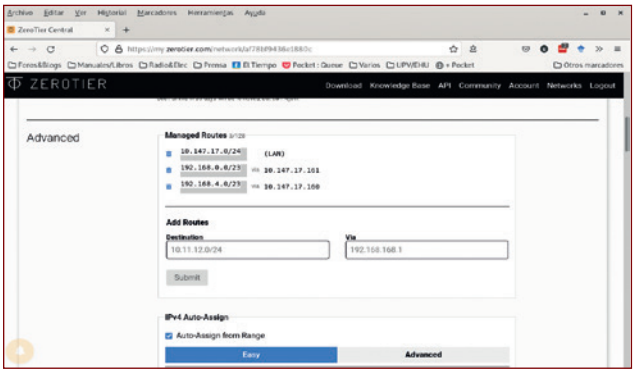

Figura 9. Especificación de rutas en ZeroTier para unir los dos segmentos de red distantes

Para llegar al segmento 192.168.1.0/23 señalamos que hemos de hacerlo vía 10.147.17.161, que es la dirección que en la VPN tiene asignada `raspi01`. Análogamente, para llegar al segmento 192.168.5.0/23 lo haremos vía `oran01`, que en la VPN tiene asignada la dirección `10.147.17.160.` (La razón por la que empleamos una máscara de 23 bits y las rutas aparecen como `192.168.0.0/23` y `192.168.4.0/23` se explica en los complementos *online* de este artículo.)

Obsérvese que las rutas definidas en ZeroTier se aplican a las máquinas en la VPN, pero no a las demás: `oran01` "sabe" que para enviar un paquete al segmento 192.168.1.0 lo ha de hacer vía `raspi01`, pero este conocimiento no alcanza al RemoteRig con dirección 192.168.5.100. Nos resta un último paso.

Paso 5: establecimiento de rutas en los dos segmentos locales

Para que la mitad del RemoteRig en 192.168.1.34 encuentre a su homóloga en 192.168.5.100 y viceversa, debemos informar a ambas del camino que deben seguir los paquetes. Podemos hacerlo en las respectivas páginas de configuración, especificando una *gateway* (que será en cada segmento la dirección IP de la máquina encargada del enrutamiento); pero podemos también hacerlo en los dos *routers* 4G, lo que hará que la ruta esté disponible para cualquier máquina que se cuelgue del segmento respectivo.

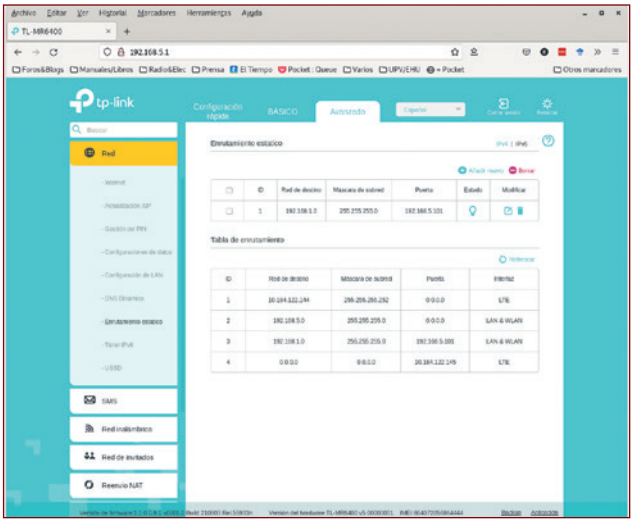

Figura 10. Especificación de ruta en el router 4G del monte. Todo el tráfico para 192.168.1.x, ha de enviarse a través de 192.168.5.101. Una ruta recíproca (a 192.168.5.x a través de 192.168.1.161) ha de especificarse en el router 4G sirviendo el segmento sede

La figura 10 muestra la ruta que necesitamos añadida al *router* 4G del segmento "monte". La modificación se hace fácilmente con el icono "Añadir nuevo" que se ve en la figura. En la mitad "sede" necesitaríamos una modificación análoga, señalando a raspi01 (en `192.168.1.161`) como puerta para acceder a la red de destino `192.168.5.0.`

Una vez hecho esto hemos completado nuestro objetivo de establecer una conexión que salta sobre el CG-NAT, y no entre dos máquinas, sino entre dos segmentos de red. Toda máquina que se conecte en uno puede transparentemente acceder a toda máquina que se conecte en otro, lo que incluye a las dos unidades RemoteRig.

Podemos verificar fácilmente que las rutas son correctas. Si desde una máquina de prueba conectada al segmento "sede" preguntamos (con `traceroute`) por la ruta hasta el RemoteRig en el segmento "monte", obtendremos una salida como la de la figura 11. Obsérvese que la máquina de prueba (`zarandoa`) no necesita formar parte de la VPN. Podemos trazar el camino inverso desde una máquina conectada en el otro segmento, verificar que tenemos conectividad a todos los equipos con herramientas como `ping`, etc.

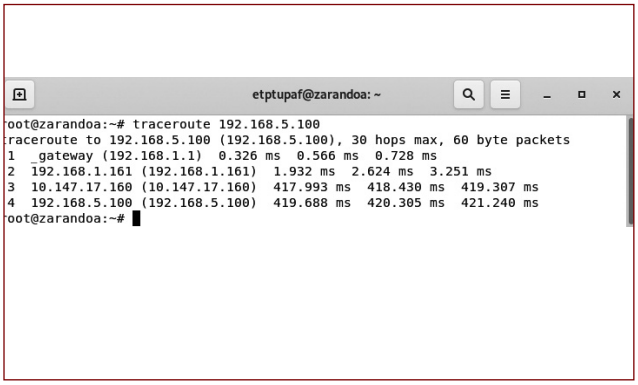

Figura 11. Camino desde una máquina en el segmento de la sede hasta el RemoteRig en el monte. El grueso del tiempo de trayecto se consume entre raspi01 (192.168.1.161) y oran01 (10.147.17.160)

Llegados a este punto, las dos mitades del RemoteRig se verán entre sí y podremos comenzar a operar. Nótese que *no necesitamos abrir puertos en ninguno de los routers, no necesitamos hacer uso de ningún servicio DNS, no necesitamos nada más.* Ambas mitades del RemoteRig trabajan como si tuvieran acceso cableado a la otra mitad. Nuestro objetivo se ha logrado.●