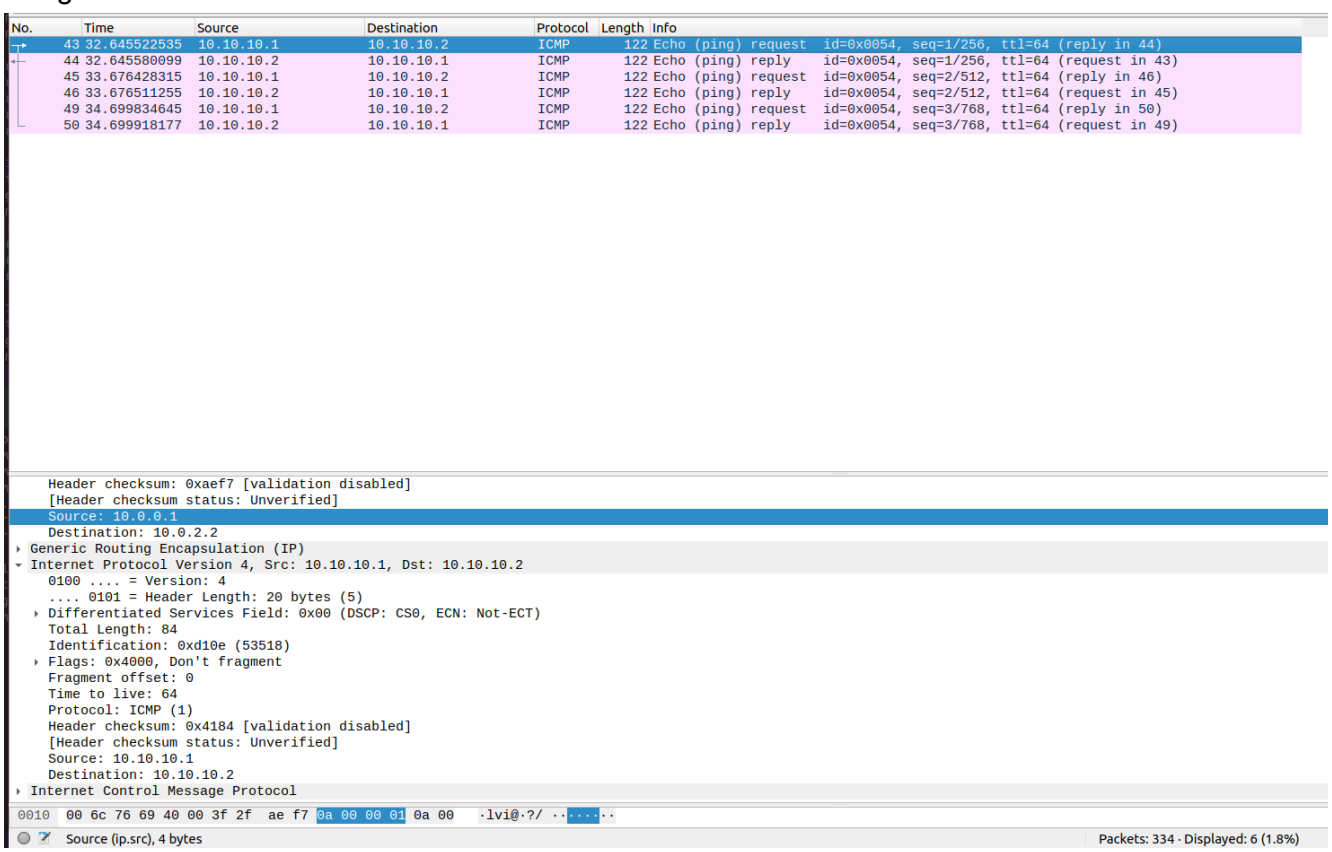
Q1:
The router that could allow traffic into home network has some NAT rules, which use a mask for packets coming in or out of that network, making the home network IPs inaccessible for exterior networks, like office. This happens because all home network IPs are masked once they pass the router and there needs to be a translation also for packets coming into that network.

Q2:
ip tunnel add gre1 mode gre local 10.0.2.2 remote 10.0.0.1
ip link set gre1 up
ip addr add 10.10.10.2/24 dev gre1

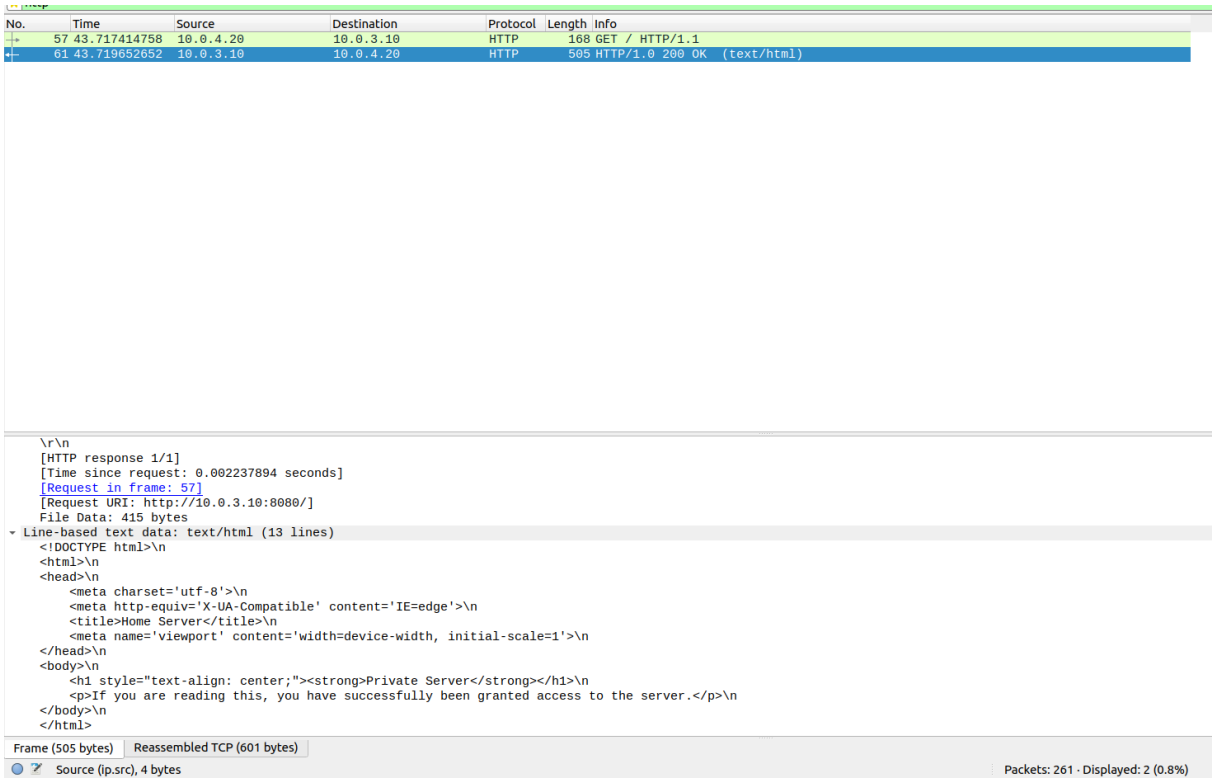
Q3:
We see in Wireshark that the ICMP packets have 2 "Internet Procotol" sections, each with a source and destination fields. In the first of these 2 sections, we have source = 10.0.0.1 and dest = 10.0.2.2, meaning that r3 will look at these fields and redirect the packet to r4. In r4, the packets are decapsulated, because it's the end of the tunnel, and r4 will look at the inner IP datagram with source = 10.10.10.1 and destination = 10.10.10.2



Q4:
ip route add 10.0.3.0/24 via 10.10.10.2

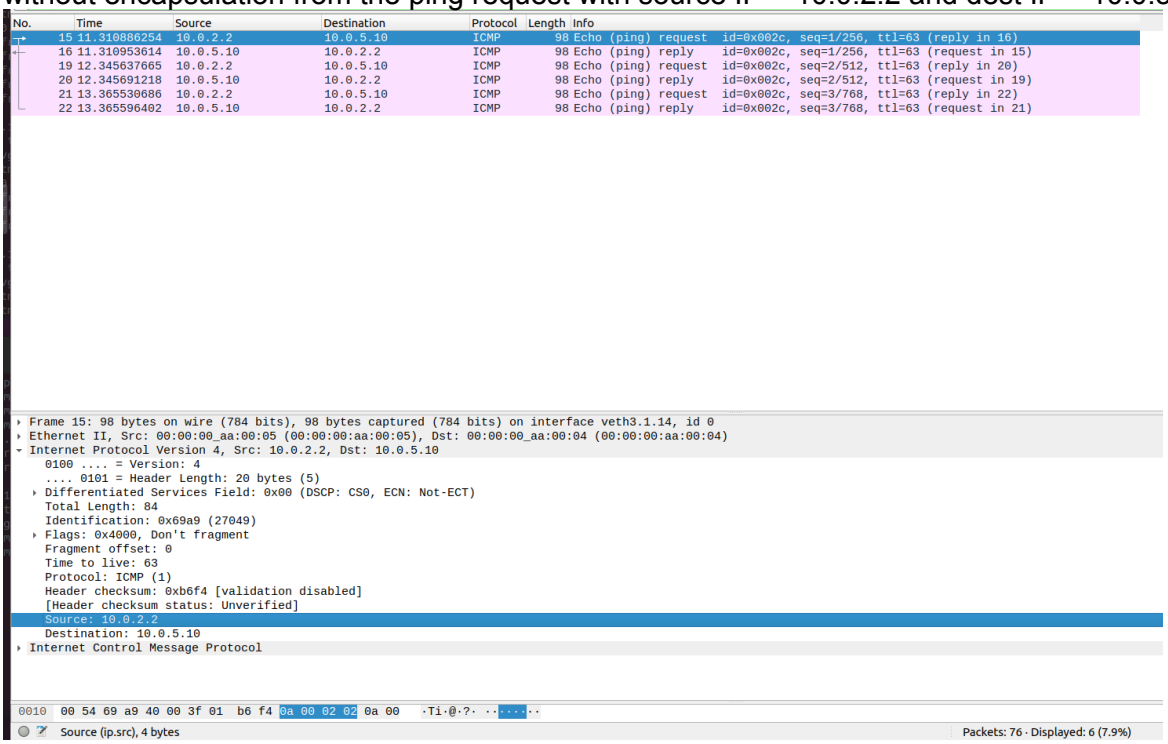
Q5:

The content is not ciphered, because we can see the content of the HTTP reply in Wireshark. If it was ciphered, we would only see the cipher and not the actual content. There is a tunnel working with encapsulation, but without ciphers.



Q6:

No, they will not be GRE-encapsulated, because it doesn't match the addresses needed to use the tunnel (encapsulation). Using Wireshark on eth1 of r3, we can see that r3 received 1 packet without encapsulation from the ping request with source IP = 10.0.2.2 and dest IP = 10.0.5.10



Q7:

It can't reach the desktop at home, because home has a VPNserver that filters the traffic, making it impossible for the packet to reach it. And there's also a masquerade present on NAT rules (same as in Q1).

Q8:

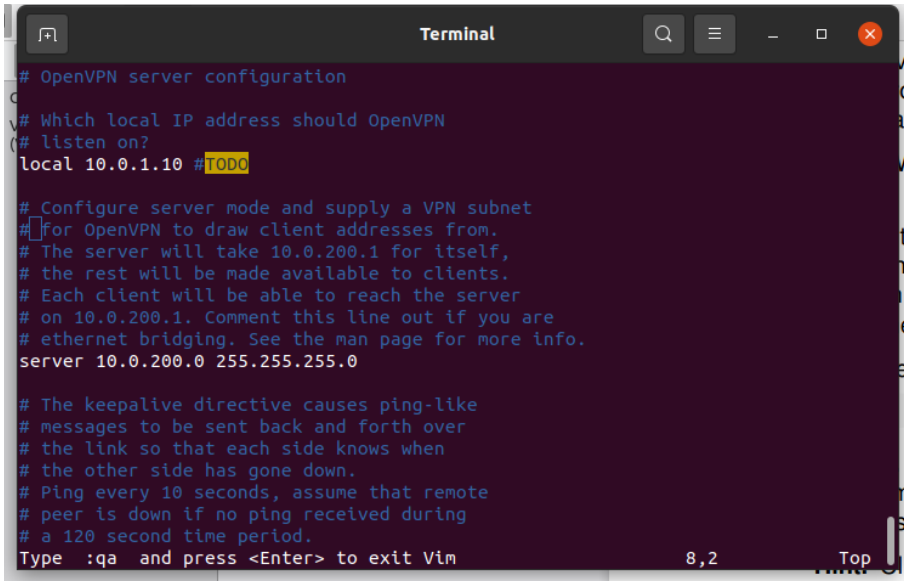
The IP where to listen is missing. The port of the server and the protocol too.

```
Terminal
# OpenVPN server configuration

# Which local IP address should OpenVPN
# listen on?
local 10.0.1.10 #TODO

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.0.200.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.0.200.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.0.200.0 255.255.255.0

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ca.crt
cert server.crt
key server.key

# Diffie hellman parameters.
dh dh.pem

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC
                                                    12,10         Top
```

```
Terminal
# Diffie hellman parameters.
dh dh.pem

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
cipher AES-256-CBC

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
log server.log

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194 #TODO

# TCP or UDP server?
proto udp #TODO

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
dev tun

# Set the appropriate level of log
# file verbosity.

# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 6
                                                    92,1          Bot
```

Q9:

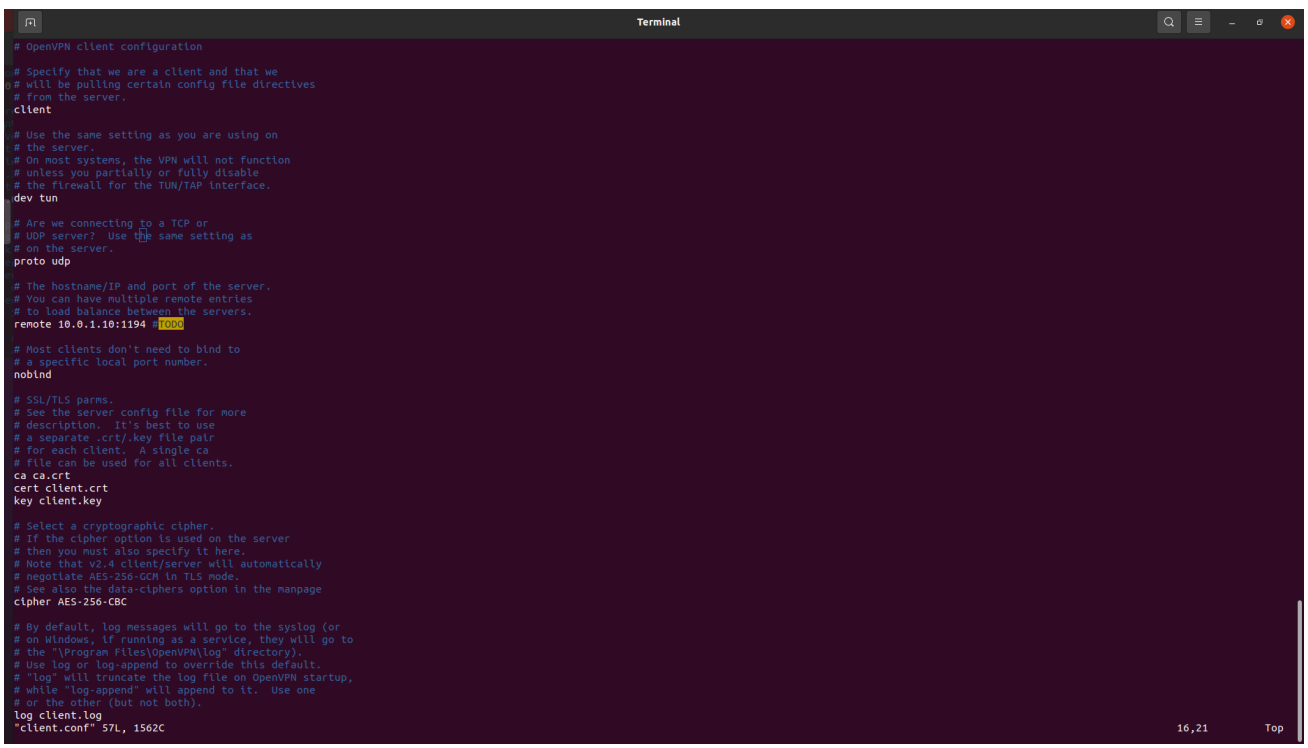
The server will get the IP 10.0.200.1



```
# OpenVPN server configuration

# Which local IP address should OpenVPN
# listen on?
local 10.0.1.10 #TODO

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.0.200.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.0.200.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.0.200.0 255.255.255.0

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
Type  :qa  and press <Enter> to exit Vim                    8,2          Top
```

Q10:

The remote IP of the OpenVPN server is missing. The ":1194" part was removed after the screenshot was taken, so it should not be considered for the answer.



```
# OpenVPN client configuration

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
dev tun

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote 10.0.1.10:1194 #TODO

# Most clients don't need to bind to
# a specific local port number.
nobind

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca ca.crt
cert client.crt
key client.key

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the data-ciphers option in the manpage
cipher AES-256-CBC

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
log client.log
"client.conf" 57L, 1562C                                    16,21         Top
```

Q11:

Using ifconfig on client we found out it's IP address on the VPN is 10.0.200.6

```
root@laptop:/tmp/pycore.40361/laptop.conf# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.20  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::200:ff:feaa:9  prefixlen 64  scopeid 0x20<link>
        inet6 2001::20  prefixlen 64  scopeid 0x0<global>
        ether 00:00:00:aa:00:09  txqueuelen 1000  (Ethernet)
        RX packets 2128  bytes 176066 (176.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 150  bytes 15206 (15.2 KB)
        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 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.0.200.6  netmask 255.255.255.255  destination 10.0.200.5
        inet6 fe80::9476:2c39:61b:7655  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 100  (UNSPEC)
        RX packets 22  bytes 1848 (1.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 2232 (2.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@laptop:/tmp/pycore.40361/laptop.conf#
```

Q12:

No VPNserver: ip route add 10.0.0.0/24 via 10.0.200.1

No client: ip route add 10.0.6.0/24 via 10.0.200.6

Q13:

No, we cannot see the contents of the HTML page, because the content is ciphered this time.

We also don't see any HTTP packet in the Wireshark screenshot, only openVPN packets.