1 Addressing exercises (without computer assistance)

1. For the following IPv4 addresses, please specify the netmask, the prefix network (netid) and the interface identifier (hostid):

2. For each of the following networks, provide the addresses IPv4 of the first host, the last one and the broadcast:
   - 192.33.182.0 /24; 10.0.0.0 /16; 81.188.160.128 /26; 81.188.160.0 /19.

3. I have the CIDR block 150.44.0.0/16 for my business. I have more than 1000 machines per subnet. Which subnetting should I use to maximize the number of subnets. For the 1st, the 2nd and the last subnet, specify the prefix, the addresses of the first machine, the last one and the broadcast.

4. What is the address space loss related to the subnetting (number of usable addresses for interfaces)?

5. I have 20 networks of 50 hosts to address. What CIDR block size shall I ask to my service provider? Suppose I was given the first block of adequate size from the prefix 60.44.32.0 /20. For the 1st, the 2nd and the last subnet, specify the prefix, the addresses of the first machine, the last one and the broadcast.

6. In the previous exercise, if we directly indicates the number of hosts that you wish to address to your provider, it will give us there a CIDR block size chosen above?

2 Addressing plan and subnetting (without computer assistance)

A company wants to integrate its network in the TCP/IP environment. It have a central site with 6 networks of 50 hosts maximum. It also wants to integrate its three branches, each with maximum 20 machines.

1. As part of an addressing based on CIDR, what scheme do you propose for the main site? Illustrate your solution a figure where you will use the first address block prefix of 88.5.100.0.

2. We integrate the three branches. Knowing that they are each connected by a dedicated leased line between their router and the one of the the main site, extend your solution and complete the figure.

3. What will be the routing table of the central router? Of the branch office router? Of a host?

4. For economy reasons, the company decided to abandon the dedicated lines and connects directly the branches to the Internet. What changes it introduced in the addressing, routing and security?

5. Finally, the company uses a private address and VPN. What is its motivation? What changes it introduced at addressing, routing, security and internet access?
3 Implementation of subnetworks on the testbed

3.1 Introduction of CISCO equipments

Cisco uses IOS (Internetwork Operating System) software in most of its equipments. This realizes the integration of interface configuration, routing and switching, thanks to a proprietary operating system of relatively old design (monolithic multitask non-preemptive). More sophisticated versions exist for high-end hardware. The interface is textual and based on command line (CLI, Command Line Interface), similar the UNIX shells, even if it does not reach the versatility of the latter. Determines a set of commands with multiple parameters is proposed according to mode and privilege level of the user (all orders have a privilege level, from 0 to 15, and will only be performed by a user with sufficient privileges).

3.1.1 The different IOS access modes

CISCO equipements use several modes in which a user can be. We use mainly the two following:

user EXEC (level 1) initial mode in which the user is located by default at the connection to the equipment. Commands available are very limited (no configuration). The prompt is of the form “equipment_name>”. To exit this mode, use the logout command or quit command.

privilege EXEC (level 15) access via the enable command. A password is required to access this mode for users with privileges (similar to the root account on UNIX). Many commands that become available as the one to enter the configuration sub-mode. The prompt is of the form “equipment_name#”. To exit this mode, use the disable command.

global configuration access via the configure command from privilege EXEC mode. In the level with privileges, this sub-mode allow to configure functionalities having a general scope of the equipment. Prompting is of the form “equipment_name(config)#”. To exit this sub-mode, use the exit command, the command end or press \[\text{ctrl} + \text{Z}\].

dedicated configuration access via commands like interface, vlan, router... from the sub-mode global configuration. Always in the level with privileges, this sub-sub-mode is used to configure specific parameters to an interface, a VLAN, routing... The prompt is of the form “equipment_name(config-xxx)#” where xxx depends on the element to configure. To exit this sub-sub-mode and return to the sub-mode of global configuration, use the exit command. To return directly to privilege EXEC mode, use the end command or press \[\text{ctrl} + \text{Z}\].

Here the sequencing of different modes cited above:

\[
\text{user EXEC} \iff \text{privilege EXEC} \iff \text{global configuration} \iff \text{dedicated configuration}
\]

3.1.2 Access to the testbed switch and router

Switches and routers present in both racks are directly accessible via TELNET:

- the management network (VLAN 200, IPv4 address 10.0./16) from the ARI host where you are logged in;
- the experimental network (VLAN N1, VLAN N2...) from your testbed VMs.

IPv4 addresses you can use to connect to the testbed equipments via the management network are presented in the Figure 1. Also, the ARI host N can access directly to 5 testbed equipments:

- the switch N, IPv4 address 10.0.2.N
- the router N, IPv4 address 10.0.3.N
- the VMs “client” (N1), “probe” N2 and “server” (N3) at the former addresses.

Remark: a password is required for equipment login, it will be given by your tutor.
3.1.3 Interface naming of testbed CISCO equipments

Switches and routers interfaces of the testbed are 100BaseTX (Fast Ethernet) type. The name of the interfaces explicitly indicate their type. Here, we use the prefix “FastEthernet” (or “fa” for short). Two numbers separated by “/” follow to differentiate the same type interfaces. Equipments can integrate several expansion cards, also the first number indicates the card and the second number the interface on the card (on the testbed, there is no additional card, then the first number is always "0"). Following to the integration of several product lines, CISCO start numbering differently interfaces on switches or on routers:

- From 1 on the switches (ie from 1 to 12 on the testbed). Eg.: “fa0/3”
- From 0 on the routers (ie 0 and 1 on the testbed). Eg.: “fa0/0”; “fa0/1”

3.1.4 General remarks for IOS commands

Table 1 presents some classical commands that are valid for the CISCO switches and the CISCO routers.

<table>
<thead>
<tr>
<th>Action</th>
<th>Commands</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>ping an IP address</td>
<td>ping &lt;ip-address&gt;</td>
<td>user</td>
</tr>
<tr>
<td>enter privilege mode</td>
<td>enable</td>
<td>user</td>
</tr>
<tr>
<td>display the running configuration</td>
<td>show running-config</td>
<td>privilege</td>
</tr>
<tr>
<td>display the interfaces</td>
<td>show interface status</td>
<td>privilege</td>
</tr>
<tr>
<td>display one interface</td>
<td>show interface fa0/&lt;if-num&gt;</td>
<td>privilege</td>
</tr>
<tr>
<td>display connected users</td>
<td>show users</td>
<td>privilege</td>
</tr>
<tr>
<td>enter in configuration mode</td>
<td>configure terminal</td>
<td>privilege</td>
</tr>
<tr>
<td>copy the configuration on a PC via TFTP</td>
<td>copy running-config tftp://&lt;PCaddr&gt;/&lt;conf-name&gt;</td>
<td>privilege</td>
</tr>
</tbody>
</table>

Negation: a command or its reciprocal are not necessarily available. For example, there is the command “shutdown” to stop an interface but no one to turn on an interface. Simply use the operator “no”, which realizes the negation of a command. Thus, “no shutdown” show a turned on interface.

Help: the two main helps that are provided by the IOS are the [tabulation] and the [?] keys. The first provides automatic completion and the second displays all choices of commands or parameters available at any time, and a short explanatory sentence.

For example, [?] taped in the beginning of line list all available commands in the current mode, and [?] taped after to the show command will list all the parameters that can follow this command.
3.2 Configuration of the switch

To realize the topology of this section, we will use two networks of experimentation. As part of the platform, we have only one switch, so we will set up these networks using VLANs.

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment. Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router or bridge. Because a VLAN is considered a separate logical network, it contains its own bridge Management Information Base (MIB) information and can support its own implementation of spanning tree.

The switches of the testbed are CISCO Catalyst 2950-12 that run the standard software image (SI) and support 64 VLANs. VLANs are identified with a number from 1 to 1005 when the SI is installed. VLAN IDs 1002 through 1005 are reserved for Token Ring and FDDI VLANs. By default all interfaces belong to VLAN 1. Two modes related to VLAN interfaces are possible:

mode “access” An access port belongs to and carries the traffic of only one VLAN. Traffic is received and sent in native formats with no VLAN tagging. Traffic arriving on an access port is assumed to belong to the VLAN assigned to the port. If an access port receives a tagged packet (802.1Q tagged), the packet is dropped, the source address is not learned, and the frame is counted in the No destination statistic. An access port can forward a tagged packet (802.1P and 802.1Q). Only one types of access ports are supported: Static access ports are manually assigned to a VLAN;

mode “trunk” A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. Only IEEE 802.1Q trunk ports are supported. An IEEE 802.1Q trunk port supports simultaneous tagged and untagged traffic. Although by default, a trunk port is a member of every VLAN known to the switch, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 1005 when the standard software image [SI] is installed) are in the allowed list.

3.2.1 Basic operation of the switch

The main commands you’ll need to configure the switch to realize the topology of the lab are presented in Table 2.

Table 2: CISCO switch commands

<table>
<thead>
<tr>
<th>Action</th>
<th>Commands</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>display the configuration of all VLANs</td>
<td>show vlan</td>
<td>privilege</td>
</tr>
<tr>
<td>summarize VLANs information</td>
<td>show vlan brief</td>
<td>privilege</td>
</tr>
<tr>
<td>create a VLAN</td>
<td>vlan &lt;vlan-num&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>name a VLAN</td>
<td>name &lt;vlan-name&gt;</td>
<td>vlan conf.</td>
</tr>
<tr>
<td>configure a VLAN</td>
<td>interface vlan &lt;vlan-num&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>associate an IP address to a VLAN</td>
<td>ip address &lt;ip-addr&gt; &lt;mask&gt;</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>remove a VLAN</td>
<td>no vlan &lt;vlan-num&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>configure an interface</td>
<td>interface fa0/&lt;if-num&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>configure several interfaces</td>
<td>interface range fa0/&lt;ifn1&gt; - &lt;ifn2&gt;, 0/&lt;ifn3&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>associate an interface to a VLAN in “access” mode</td>
<td>switchport access vlan &lt;vlan-num&gt;</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>associate an interface to the VLAN aggregate (mode “trunk”)</td>
<td>switchport mode trunk</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>restrict the aggregate by removing/adding VLANs (mode “trunk”)</td>
<td>switchport trunk allowed vlan remove &lt;vlan-num&gt;, ou switchport trunk allowed vlan add &lt;vlan-num&gt;</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>display the configuration of an interface</td>
<td>show interfaces fa0/&lt;if-num&gt; switchport</td>
<td>privilege</td>
</tr>
<tr>
<td>display the configuration of the interface associated to the aggregate (mode “trunk”)</td>
<td>show interfaces fa0/&lt;if-num&gt; trunk</td>
<td>privilege</td>
</tr>
</tbody>
</table>
3.2.2 Switches interfaces of the testbed

The switches of the testbed have 12 Ethernet interfaces. Associated cabling to each of these interfaces is fixed. See the Table 3 for the numbering of the interfaces (references associated to each interfaces are displayed for the switch N, corresponding to the ARI host N).

Table 3: CISCO Catalyst 2950-12 interfaces

<table>
<thead>
<tr>
<th>Interface number</th>
<th>IOS name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>fa0/1</td>
<td>connected to the fa0/0 router N interface</td>
</tr>
<tr>
<td>2</td>
<td>fa0/2</td>
<td>connected to the fa0/1 router N interface</td>
</tr>
<tr>
<td>3</td>
<td>fa0/3</td>
<td>connected to the eth1 interface of the &quot;client&quot; VM N1</td>
</tr>
<tr>
<td>4</td>
<td>fa0/4</td>
<td>connected to the eth1 interface of the &quot;sniffer&quot; VM N2</td>
</tr>
<tr>
<td>5</td>
<td>fa0/5</td>
<td>connected to the eth1 interface of the &quot;server&quot; VM N3</td>
</tr>
<tr>
<td>6</td>
<td>fa0/6</td>
<td>not used in this lab</td>
</tr>
<tr>
<td>7</td>
<td>fa0/7</td>
<td>not used in this lab</td>
</tr>
<tr>
<td>8</td>
<td>fa0/8</td>
<td>not used in this lab</td>
</tr>
<tr>
<td>9</td>
<td>fa0/9</td>
<td>not used in this lab</td>
</tr>
<tr>
<td>10</td>
<td>fa0/10</td>
<td>not used in this lab</td>
</tr>
<tr>
<td>11</td>
<td>fa0/11</td>
<td>connected to the main switch of the rack</td>
</tr>
<tr>
<td>12</td>
<td>fa0/12</td>
<td>connected to the ARI N host</td>
</tr>
</tbody>
</table>

1. From the ARI host N, log onto the corresponding switch via telnet (telnet 10.0.2.N).

2. View the switch configuration, especially the parameters of the interfaces and VLANs (these studied VLANs may be already created and with numbers N1, N2, 200 and respective names VLAN_A, VLAN_B, VLAN_ADMIN).

3.2.3 Interface mirroring (SPAN)

A switch generally directly forward frames from an input port to an output port. The interception of traffic from another port is not feasible as on a simple hub. The realization of a capture from a probing machine different from the client or server, i.e. from an interface different from that of the client or server, requires the use of a specific frame copy mechanisms to the interface of the probe.

The frame copy mechanism, also called “port mirroring”, “port monitoring” or “SPAN” (Switched Port Analyzer in CISCO) allow traffic replication. Its principle is as follows: all frames arriving on an interface is duplicated to a other interface specified during configuration. As part of the platform, we realize the copying of frames from by one of the interfaces connected to the client or server machines to the interface connected to the machine ‘probe. Table 4 present few IOS commands for ‘has this technique.

Table 4: Commands for interface mirroring (SPAN)

<table>
<thead>
<tr>
<th>Action</th>
<th>Commandes</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>specify the source interface</td>
<td>monitor session 1 source interface fa0/&lt;num-if&gt; [ , , -]</td>
<td>global conf.</td>
</tr>
<tr>
<td>specify the destination interface</td>
<td>monitor session 1 destination interface fa0/&lt;num-if&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>display mirroring session</td>
<td>show monitor</td>
<td>global conf.</td>
</tr>
<tr>
<td>remove mirroring session</td>
<td>no monitor session all</td>
<td>global conf.</td>
</tr>
</tbody>
</table>

To implement the interface mirroring on the testbed, a SPAN session is used. This corresponds to the association of a destination interface with multiple source interfaces (more sophisticated switches than the CISCO Catalyst 2950-12 may use VLANs sources but we will only describe in this paper the relatively restricted functionalities our testbed switches).
A **source port**, also known as “monitored port”, is an switched interface which we wishes to analyze the related traffic. In a single session, the traffic received (Rx), the traffic emis (Tx) or bidirectional traffic (Both) may be mirrored. By default, bidirectional traffic from source interface is overseeing.

Any local SPAN session must be associated to a **destination port**, also called “monitoring port”. By default, traffic that is copy is identical to that of the source interfaces supervised. We will not use any specific encapsulation of the mirrored traffic (no encapsulation type header).

1. Display the configuration of the copying of switch interfaces and justify it.
2. What IOS commands allowed to set up the session mirroring interfaces?

### 3.2.4 Setup of 2 VLANs

With the first topology of the testbed (used in the labs 2 to 5), we transmit all traffic between the “client” and “server” VMs directly through the switch. For this, the interfaces linking the “client” and “server” VMs to the switch are placed within the same VLAN (see Figure 2).

![Figure 2: Virtual topology 1 (1 LAN)](image)

1. What IOS commands helped set up the VLAN N1?

In the first topology, interfaces of the VMs were on the same VLAN. There was no need for routing and therefore not router configuration. We want to implement the second topology, using two VLANs to separate the “client” and the “server” VMs. The purpose of this modification is to study the inter-network traffic, and especially the integration of routers (see Figure 3).

![Figure 3: Virtual topology 2 (2 LANs and 1 router)](image)

We will keep the VLAN N1 for the client and introduce a new, the VLAN N2 for the server:
2. What IOS commands allow to implement the new VLAN? Set up the new configuration if needed.
3. Verify now that the “client” and “server” VMs can not communicate directly.
4. Can you still capture the corresponding traffic using the tools wireshark on the “probe” VM?
5. Analyze traffic generates while you test the connectivity between the “client” and the “server” VMs.
3.3 Configuring the router

In topology 2, to make communicate the “client” and the “server” VMs, we need to add a router because both VMs are now in different networks. CISCO routers 2800 platform have two Ethernet interfaces. Each of these interfaces must be on one of the two VLANs to experimentation (VLAN N1 and VLAN N2).

3.3.1 Basic router commands

The main commands you’ll need to configure the router in the topology 2 are presented in the Table 5.

<table>
<thead>
<tr>
<th>Action</th>
<th>Commands</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>display the summary of the interfaces</td>
<td>show interfaces brief</td>
<td>privilege</td>
</tr>
<tr>
<td>configure an interface</td>
<td>interface fa0/&lt;num-if&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>configure a virtual interface</td>
<td>interface fa0/&lt;num-if&gt;.&lt;num-sub&gt;</td>
<td>global conf.</td>
</tr>
<tr>
<td>associate an IPv4 address to an interface</td>
<td>ip address &lt;adresse-ip&gt; &lt;masque&gt;</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>do not associate IPv4 address to an interface</td>
<td>no ip address</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>remove the proxy ARP interface</td>
<td>no ip proxy-arp</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>selection of a VLAN interface in mode “trunk”</td>
<td>encapsulation dot1Q &lt;vlan-num&gt;</td>
<td>interf. conf.</td>
</tr>
<tr>
<td>display the routing table</td>
<td>show ip route</td>
<td>privilege</td>
</tr>
</tbody>
</table>

1. What configuration should be set up at the switch to integrate the router?
2. What configuration should be set up at the router? What IOS commands should be used to realize it?
3. Which routing protocol should we start to build the routing table? How to display this last?
4. From the ARI host N, log onto the corresponding router via via telnet (telnet 10.0.3.N).
5. Visualize and analyze the configuration of the router.

3.3.2 Configuring virtual interfaces

Virtual interfaces are network interfaces that are not a physical interface. Physical interfaces have some form of physical element, for example, an RJ-45 male connector on an Ethernet cable. Virtual interfaces exist only in software; there are no physical elements.

Subinterfaces are associated with physical interfaces. Subinterfaces are enabled when the physical interface with which they are associated is enabled, and subinterfaces are disabled when the physical interface is shut down. Subinterfaces are created by subdividing the physical interface into two or more virtual interfaces on which you can assign unique Layer 3 network addresses such as IP subnets, so that the IP routing protocols would see the network connection to each remote networking device as a separate physical interface even though the subinterfaces share a common physical interface. Subinterfaces are identified by a prefix that consists of the hardware interface descriptor followed by a period and then by a number that is unique for that prefix. The full subinterface number must be unique to the networking device. For example, the first subinterface for Ethernet interface fa0/1 might be named fa0/1.1 where .1 indicates the subinterface.

The router configuration shows the use of subinterfaces (fa0/1.1 and fa0/1.2). These subinterfaces are necessary because the topology 2 (see Figure 3) uses 3 interfaces and the router provides 2 physical interfaces.

The router is directly connected to networks 10.0/16, 10.N.1/24 and 10.N.2/24. The interface (fa0/1) is subdivided declaring it functional but with no IP address (no ip address command). subinterfaces fa0/1.1 and fa0/1.2 are declared in a similar way as physical interfaces.

The physical interface of the router being linked to a switch interface in “trunk” mode, all VLANs are available. It remains that in associating each of these subinterfaces to a corresponding VLAN (encapsulation dot1Q command with the number of the corresponding VLAN).
3.3.3 Test communication between 2 VLANs

After having verified the parameters of the interfaces (fa0/1.1 and fa0/1.2), be sure to delete the proxy ARP functionality configured by default on each interface on the testbed routers. This feature allows the router to answer to ARP requests from misconfigured host. It could interfere with the diagnoses that we realize:

```
routeurN# configure terminal
routeurN(config)# interface fa0/1.1
routeurN(config-if)# no ip proxy-arp
routeurN(config)# interface fa0/1.2
routeurN(config-if)# no ip proxy-arp
routeurN(config)# end
```

Once the router configuration adequate, routing between both subnets is operational.

1. Check the connectivity from the router (using the ping to the client to the server).
2. Check the connectivity to the router from the “client” and “server” VMs (still using the ping command).
3. Could the “client” and “server” VMs communicate with each other again?

3.3.4 Reconfiguring the testbed VMs

Changing network parameters involves modifying the configuration of all equipments directly connected to these networks. This is the case for the “client” and ‘server’ VMs. UNIX commands used are: ifconfig and route.

Here is the beginning of the description of the manpage of the UNIX command ifconfig:

```
Ifconfig is used to configure the kernel-resident network interfaces. It is used at boot time or when system tuning is needed. If no arguments are given, ifconfig displays the status of the currently active interfaces. Otherwise, it configures an interface.
```

And here is one of the route command:

```
Route manipulates the kernel’s IP routing tables. Its primary use is to set up static routes to specific hosts or networks via an interface after it has been configured with the ifconfig(8) program. When the add or del options are used, route modifies the routing tables. Without these options, route displays the current contents of the routing tables.
```

When used to modify the network configuration of the machine, these commands require privileges administrator and then use the su command.

1. What analysis using wireshark can you do about the situation?
2. Check your diagnosis of the fault using /sbin/ifconfig on the VMs (on the eth1 interfaces on the client and the server).
3. A modification of the mask of the interfaces of used VMs is necessary. Use the following command: /sbin/ifconfig eth<n> netmask <nouveau-masque>.
4. Could the “client” and “server” VMs communicate with each other again?
5. Does an analysis using wireshark can help you diagnose the situation?
6. An indication of the gateways to access networks that are not directly connected is necessary. Use the following command: /sbin/route add default gw <adresse-passerelle>.
7. The “client” and “server” VMs can communicate with each other last?
3.4 Before leaving the room ...

At the end of the lab (not right away! You still need the configuration done), remember, if your tutor ask for it, to put the testbed machines in their initial state (the one where you have found them):

- delete the VLAN N2 from the switch (telnet 10.0.2.N):

  ```
  swN#show vlan brief
  swN#configure terminal
  swN(config)#interface range fa0/3, fa0/5
  swN(config-if-range)#switchport access vlan N1
  swN(config-if-range)#exit
  swN(config)#no vlan N2
  swN(config)#end
  swN#show vlan brief
  ```

- remove potential modifications made to the configuration of the router (telnet 10.0.254.N).

- reconfigure each “client” and “server” VMs ...
  - VM “client” (telnet 10.0.7.N1):
    ```
    client@vmN1:~$ su
    root@vmN1:~$ /sbin/ifconfig eth0 netmask 255.255.0.0
    root@vmN1:~$ /sbin/route del default
    ```
  - VM “serveur” (telnet 10.0.7.N3):
    ```
    client@vmN3:~$ su
    root@vmN3:~$ /sbin/ifconfig eth2 netmask 255.255.0.0
    root@vmN3:~$ /sbin/route del default
    ```

4 ICMP control messages

4.1 Study of the ping tool

Here is the beginning of the description of the manpage on ping command:

Ping uses the ICMP protocol’s mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (‘pings’) have an IP and ICMP header, followed by a struct timeval and then an arbitrary number of ‘pad’ bytes used to fill out the packet.

4.2 Testing a remote machine

The ping command allow to test connectivity to a remote machine, and — by sending multiple packets to the following — to compile statistics on the characteristics of the path (RTT, loss rate, variability of results depending the size of sender datagrams...). An example of use:

```
pirogue:~# ping sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253): 56 data bytes
64 bytes from 132.227.74.253: icmp_seq=0 ttl=247 time=5.5 ms
64 bytes from 132.227.74.253: icmp_seq=1 ttl=247 time=9.3 ms
64 bytes from 132.227.74.253: icmp_seq=2 ttl=247 time=8.0 ms
64 bytes from 132.227.74.253: icmp_seq=3 ttl=247 time=6.3 ms
64 bytes from 132.227.74.253: icmp_seq=4 ttl=247 time=4.8 ms
64 bytes from 132.227.74.253: icmp_seq=5 ttl=247 time=7.6 ms
64 bytes from 132.227.74.253: icmp_seq=6 ttl=247 time=5.8 ms
--- sphinx.lip6.fr ping statistics ---
```
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 4.8/6.7/9.3 ms
pirogue:~#

1. For training, scan manually (without wireshark) the first frame corresponding to this exchange:

```
0000 00 08 21 59 66 42 00 04 76 21 1b 95 08 00 45 00 ..!YfB.. v!.....E.
0010 00 54 64 db 00 00 40 01 df 38 c2 fe a3 b6 84 e3 .Tdû..@. sR gà¶.ä
0020 4a fd 08 00 d8 2d 6e 5b 00 00 3f 9f 10 01 00 0d Jý."Q-"?......
0030 76 c6 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 v£....... ........
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ........ ... !"#$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 k'( )*+- ./.012345
0060 36 37 67
```

2. In your opinion, What do the data transported by the ICMP message previously analysed? Use the manpage for complementary information on ping.

3. Realize on the testbed the capture a ping between client and server using usual software wireshark (or load the file: /Infos/lmd/2014/master/ue/ares-2014oct/tme6-pin.dmp.gz).

witch exchange could you observe in the trace?

4. Whitch answer is returned at the request analyzed previously?

5. Analyze the fields of multiple exchanged ICMP packets. Infer their meaning and their usefulness for ping.

### 4.3 Testing of a close host with recording of the path

The ping \(-R\) is a variant of the previously illustrated execution where the options are introduced in the IPv4 header. Discover it in the following example:

```
pirogue:~# ping -R rap-jussieu.cssi.renater.fr
PING rap-jussieu.cssi.renater.fr (193.51.182.201): 56 data bytes
64 bytes from 193.51.182.201: icmp_seq=0 ttl=252 time=11.2 ms
RR: pirogue.l2ti.univ-paris13.fr (194.254.163.182)
    192.168.208.252
    paris13-jussieu.cssi.renater.fr (193.51.182.205)
    jussieu-a1-0-65.cssi.renater.fr (193.51.182.202)
    rap-jussieu.cssi.renater.fr (193.51.182.201)
    jussieu-f3-3.cssi.renater.fr (193.51.182.206)
    192.168.208.254
    gw163.univ-paris13.fr (194.254.163.254)
    pirogue.l2ti.univ-paris13.fr (194.254.163.182)
```

--- rap-jussieu.cssi.renater.fr ping statistics ---
5 packets transmitted, 1 packets received, 80% packet loss
round-trip min/avg/max = 11.2/11.2/11.2 ms
pirogue:~#
1. Realize on the testbed the capture a ping -R between client and server using usual software wireshark (or load the file: /Infos/lmd/2014/master/ue/ares-2014oct/tme6-pRp.dmp.gz). What new information could you observe in the trace?

2. Analyze ICMP fields. Infer their meaning, with the contents of the IP option, their usefulness for ping -R.

3. Can you draw a figure of the networks crossed?

4. Test with a record of the path machine eloignee

The ping -R uses the options field limit of the IP header. What's going on there in the following example?

```
pirogue:~# ping -R sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253): 56 data bytes
64 bytes from 132.227.74.253: icmp_seq=0 ttl=247 time=8.8 ms
RR: pirogue.12ti.univ-paris13.fr (194.254.163.182)
    192.168.208.252
    paris13-jussieu.cssi.renater.fr (193.51.182.205)
    jussieu-al-0-65.cssi.renater.fr (193.51.182.202)
    gw-rap.rap.prdf.fr (195.221.126.78)
    rap-jussieu.rap.prdf.fr (195.221.127.181)
    r-jusren.reseau.jussieu.fr (134.157.254.126)
    r-olympo.ip6.fr (132.227.109.254)
    132.227.74.254
64 bytes from 132.227.74.253: icmp_seq=1 ttl=247 time=3099.5 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=2 ttl=247 time=2099.7 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=3 ttl=247 time=1099.9 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=4 ttl=247 time=100.1 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=5 ttl=247 time=8.6 ms (same route)
64 bytes from 132.227.74.253: icmp_seq=6 ttl=247 time=14.0 ms (same route)
--- sphinx.ip6.fr ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 8.6/918.6/3099.5 ms
```

1. Load the following file: /Infos/lmd/2014/master/ue/ares-2014oct/tme6-pRl.dmp.gz. What new information could you discover in these frames?

2. Analyze the route recorded in IP packets back. What differences do you see?

3. What limitations do you see for the ping -R approach to deduce the route taken by packets to a destination given?

5 Study of the traceroute tool

Here is the beginning of the description of the manpage on traceroute command:

```
The Internet is a large and complex aggregation of network hardware, connected together by gateways. Tracking the route one’s packets follow (or finding the miscreant gateway that’s discarding your packets) can be difficult. Traceroute utilizes the IP protocol ‘time to live’ field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host.
```

The trace route command allow to get address of all intermediate host along a path to a destination. Here is an example:

```
pirogue:"# traceroute sphinx.lip6.fr
traceroute to sphinx.lip6.fr (132.227.74.253), 30 hops max, 38 byte packets
 1 gw163.univ-paris13.fr (194.254.163.254) 2.579 ms 0.496 ms 0.415 ms
 2 192.168.208.254 (192.168.208.254) 0.640 ms 0.602 ms 0.216 ms
```
1. On the testbed, use the traceroute command between the client and the server (or download the following file: /Infos/lmd/2014/master/ue/ares-2014oct/tme6-tcr.dmp.gz).

   Analyse the first traceroute frame sender. What is its purpose?

2. Which event generate the next frame? What is the interest of such phenomenon?

3. How the traceroute command generate its answers?

4. Why the destination UDP port number are they increasing?

5. At the end, how to stop the traceroute iteration?