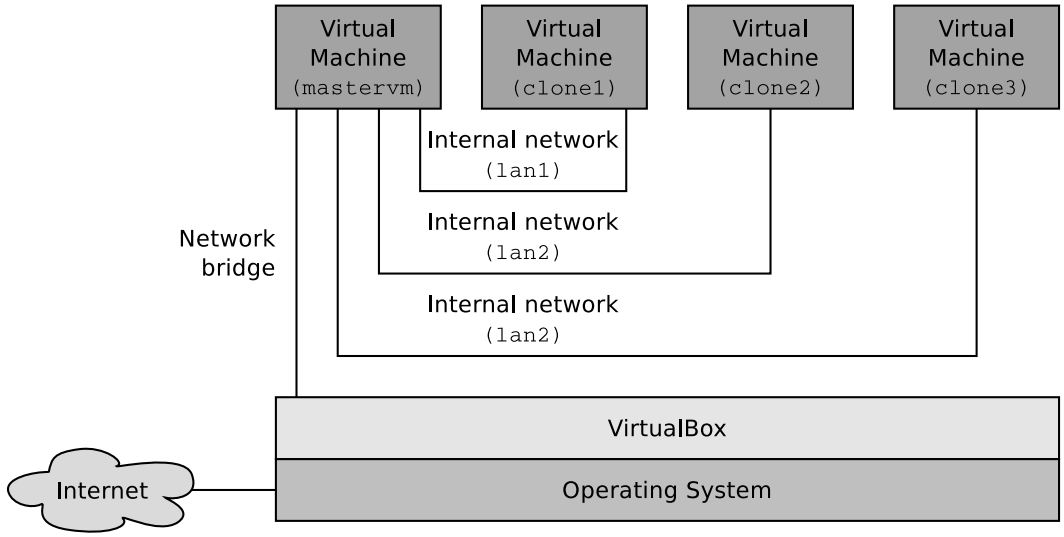
Lab Exercise Sheet 2

Document and analyze your experimental procedures by using your Wireshark and terminal recordings. Note all relevant intermediate steps. Mark and explain all relevant information, such as protocol header fields, MAC addresses, IP addresses, port numbers. If you have little experience with Linux, you may need to do some research. Send your self prepared experiment documentation in the PDF file format to christianbaun@fb2.fra-uas.de, cocos@stud.fra-uas.de and spanou@stud.fra-uas.de. Alternatively, fill out the document, print it out, and submit it during one of the exercise sessions.

First name:	Henry-Norbert
Last name:	Cocos
Student number:	Sample Solution

1. Clone the VM from the 1st lab exercise (from now on we will call this VM `mastervm`) three times with the VirtualBox user interface.

   - Specify for each new VM a unique name (e.g. `clonevm[1-3]`) and specify that each VM will get a new MAC address.
   - Add three new virtual interfaces to the `mastervm` via the VirtualBox user interface. Each one of these new network interfaces must be attached to different *internal networks*. The name of each internal network must be unique. e.g. `lan[1-3]`.
   - The network interface of each one of `clonevm[1-3]` also need to be attached to an *Internal Network* in the VirtualBox user interface. Connect `clonevm1` to the first internal network (e.g. `lan1`), `clonevm2` to the second internal network (e.g. `lan2`), and so on.



Check the relevant MAC addresses and write them into this table:

Your local Router to the internet:	34:31:c4:74:70:92
Physical network interface of your host:	4c:34:88:9e:4d:28
`mastervm` (bridged interface):	08:00:27:52:da:cd
`mastervm` (internal interface 1):	08:00:27:ff:6d:96
`mastervm` (internal interface 2):	08:00:27:33:3c:aa
`mastervm` (internal interface 3):	08:00:27:28:73:72
`clonevm1` (internal interface):	08:00:27:e8:a4:af
`clonevm2` (internal interface):	08:00:27:a1:25:5c
`clonevm3` (internal interface):	08:00:27:cb:0f:3e

The `mastervm` should operate as a Bridge/Switch between the bridged network interface and the three new network interfaces for `clonevm[1-3]`, which are attached to the internal networks `lan[1-3]`.

- Install the command line tools for bridging Ethernet connections (package `bridge-utils`) on the `mastervm`.
- You have several options to implement the IP forwarding.
  - Option 1: Create[1] a new logical Bridge with the command line tool `brctl` on the `mastervm`. Add the four virtual network interfaces of the `mastervm` to the logical Bridge.
  - Option 2: Specify for `lan[1-3]` three independent address spaces (e.g. `192.168.10.0/24`, `172.22.0.0/16` and `192.168.60.0/24`). Assign[2] valid IP addresses and further network configuration parameters to the virtual network devices inside the `mastervm` and `clonevm[1-3]`. Implement IP package forwarding (NAT-Masquerading)[3].

Copy the content of the IP routing table of the `mastervm` into this field:

```
# route -n

Kernel-IP-Routentabelle
Ziel            Router          Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.178.1   0.0.0.0         UG    0      0        0 enp0s3
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s9
192.168.3.0     0.0.0.0         255.255.255.0   U     0      0        0 enp0s10
192.168.178.0   0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
```

[1] `http://www.tldp.org/HOWTO/BRIDGE-STP-HOWTO/set-up-the-bridge.html`

[2] This can be done with command line tools like `ip` or `ifconfig` or inside the file `/etc/network/interfaces`.

[3] This can be done with command line tools like `ip` or `iptables` or inside the file `/etc/network/interfaces`.

Copy the relevant content of the file `/etc/network/interfaces` of the `mastervm` into this field:

```
# /etc/network/interfaces

auto lo
iface lo inet loopback


# WAN Interface
auto enp0s3
iface enp0s3 inet dhcp


# LAN 1
auto enp0s8
iface enp0s8 inet static
  address 192.168.1.1
  netmask 255.255.255.0
  broadcast 192.168.1.255

# LAN 2
auto enp0s9
iface enp0s9 inet static
  address 192.168.2.1
  netmask 255.255.255.0
  broadcast 192.168.2.255

# LAN 3
auto enp0s10
iface enp0s10 inet static
  address 192.168.3.1
  netmask 255.255.255.0
  broadcast 192.168.3.255

# Delete previous iptables configuration
up /sbin/iptables -F
 up /sbin/iptables -X
  up /sbin/iptables -t nat -F

# Enable NAT-Forwarding for all interfaces
up /sbin/iptables -A FORWARD -o enp0s3 -s 0.0.0.0/0 -m conntrack --ctstate NEW -j
ACCEPT
 up /sbin/iptables -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j
ACCEPT
  up /sbin/iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
   up /sbin/sysctl -w net.ipv4.ip_forward=1
```

2. Check the content of the ARP cache on the `mastervm`. Copy the content of the ARP cache of the `mastervm` into this field:

```
# arp -n

Adresse Hardware-Typ Hardware-Adresse Optionen Maske Schnittstelle
192.168.178.1           ether   34:31:c4:74:70:92   C                     enp0s3
```

Send some ping requests between the `mastervm` and `clonevm[1-3]`. Copy the content of the ARP cache of the `mastervm` into this field:

```
# arp -n

Adresse Hardware-Typ Hardware-Adresse Optionen Maske Schnittstelle
192.168.1.10            ether   08:00:27:e8:a4:af   C                     enp0s8
192.168.178.1           ether   34:31:c4:74:70:92   C                     enp0s3
192.168.3.30            ether   08:00:27:cb:0f:3e   C                     enp0s10
192.168.2.20            ether   08:00:27:a1:25:5c   C                     enp0s9
```

3. Do a ping operation from `clonevm1` to the address `debian.org`. The ping operation will cause the transmission of an ARP request and an ARP reply. Monitor these transmissions with Wireshark from the `mastervm`. Copy the relevant information (MAC addresses and IP addresses of sender and target) of the ARP request into this field:

ARP Request

Sender MAC address: (08:00:27:52:da:cd)
Sender IP address: 192.168.178.57
Target MAC address: (00:00:00:00:00:00)
Target IP address: 192.168.178.1

Source: arp_ping_debian_clone1.pdf

Copy the relevant information (MAC addresses and IP addresses of sender and target) of the ARP reply into this field:

ARP Reply

Sender MAC address: (34:31:c4:74:70:92)
Sender IP address: 192.168.178.1
Target MAC address: (08:00:27:52:da:cd)
Target IP address: 192.168.178.57

Source: arp_ping_debian_clone1.pdf

Which network protocols are involved in the transmission of the ARP messages? Assign them to the protocol stack.

Layer 7:	
Layer 6:	
Layer 5:	
Layer 4:	
Layer 3:	IPv4
Layer 2:	Ethernet / ARP
Layer 1:	Wired Connection (Ethernet)

What is the destination address of the frame, that is used to transmit the ARP request?

00:00:00:00:00:00

What is the value of the `type` header field inside the frame, that is used to transmit the ARP requrest?

ARP (0x0806)

Which IP address belongs to the sender MAC address in the header of the frame, that is used to transmit the ARP requrest?

Sender IP address: 192.168.178.57

What is the destination IP address, to which the matching MAC address is searched inside the ARP request?

Target IP address: 192.168.178.1

What is the name of the header field, that is used to store the destination IP address inside the ARP request?

Target IP address

What is the value of the header field, that is used to store the destination MAC address inside the ARP request?

Target MAC address

Expand the protocol window of the first Layer of protocols inside Wireshark, that are involved in the first echo request/reply message pair and copy the relevant information into this field:

Echo (ping) request

171/177.346662292/192.168.178.57/149.20.4.15/ICMP 98 Echo (ping) request id=0x0aba, seq=1/256, ttl=63 (reply in 172)
Frame 171: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: PcsCompu_52:da:cd (08:00:27:52:da:cd), Dst: Avm_74:70:92 (34:31:c4:74:70:92)
Internet Protocol Version 4, Src: 192.168.178.57, Dst: 149.20.4.15
Internet Control Message Protocol

Echo (ping) request

172/177.549735852/149.20.4.15/192.168.178.57/ICMP 98 Echo (ping) reply id=0x0aba, seq=1/256, ttl=53 (request in 171)
Frame 172: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
Ethernet II, Src: Avm_74:70:92 (34:31:c4:74:70:92), Dst: PcsCompu_52:da:cd (08:00:27:52:da:cd)
Internet Protocol Version 4, Src: 149.20.4.15, Dst: 192.168.178.57
Internet Control Message Protocol

Source: ping_debian_clone1_Layer1.pdf

Which network protocols are involved in the transmission of the echo request/reply message pair? Assign them to the protocol stack.

Layer 7:	
Layer 6:	
Layer 5:	
Layer 4:	
Layer 3:	IPv4 / ICMP
Layer 2:	Ethernet
Layer 1:	Wired Connection (Ethernet)

What is the length of the IP header in bytes?

20 Bytes

What is the length of the ICMP payload in bytes?

48 Bytes

To which destination IP address is the ICMP request sent?

149.20.4.15

What is the matching MAC address?

34:31:c4:74:70:92

What is the value of the `type` header field inside the frame, that is used to transmit the ICMP request?

8 (Echo (ping) request)

What is the purpose of the sequence number inside the ICMP header?

The Sequence Number is used to match each reply to its corresponding request