

Administration Système Linux

Projet 1

Adressage et Routage

M1 Computer Science Network

Système réseau



UNIVERSITÉ ÉVRY
PARIS-SACLAY

Nom: KOKPATA

Prénom : Eudes

N°Étudiant : 20220808

Université : Paris-Saclay

Enseignant : Pascal PETIT

Sommaire

1 Configuration des machines A, B et C.....	3
1.1 Maquette du réseau :.....	3
1.1.1 Machine A.....	3
1.1.2 Machine B	4
1.1.3 Machine C.....	4
2 Questions Réponses.....	5
2.1 Question 1	5
2.1.1 Réponse 1.....	5
2.2 Question 2	5
2.2.1 Réponse 2.....	5
2.3 Question 3	7
2.3.1 Réponse 3.....	7
2.4 Question 4.....	7
2.4.1 Réponse 4.....	7
3 Trajet d'un paquet « ping » de A à C : IP, MAC, TTL.....	8
3.1.1 Analyse du Processus de Routage et des En-têtes de Paquet.....	8
3.1.2 Décision de Routage et Transfert.....	8
3.1.3 Identification des Paquets.....	8
3.1.4 Time to live (TTL).....	8
3.1.5 Modification des En-têtes lors du Routage.....	8
3.2 Les Paquets.....	10
3.2.1 Ping de A vers C sur R1.....	10
3.2.2 Ping de A vers C sur R2.....	10
3.2.3 SSH de A vers C sur R1.....	11
3.2.4 SSH de A vers C sur R2.....	11

Introduction

Dans le cadre de ce projet d'administration système sous Linux (Debian 12), nous allons mettre en place une maquette réseau constituée de trois machines virtuelles. L'objectif est de comprendre et de manipuler la configuration réseau de machines sous Linux en utilisant deux types de réseaux différents : un réseau NAT et un réseau Host-Only.

La maquette repose sur l'architecture suivante :

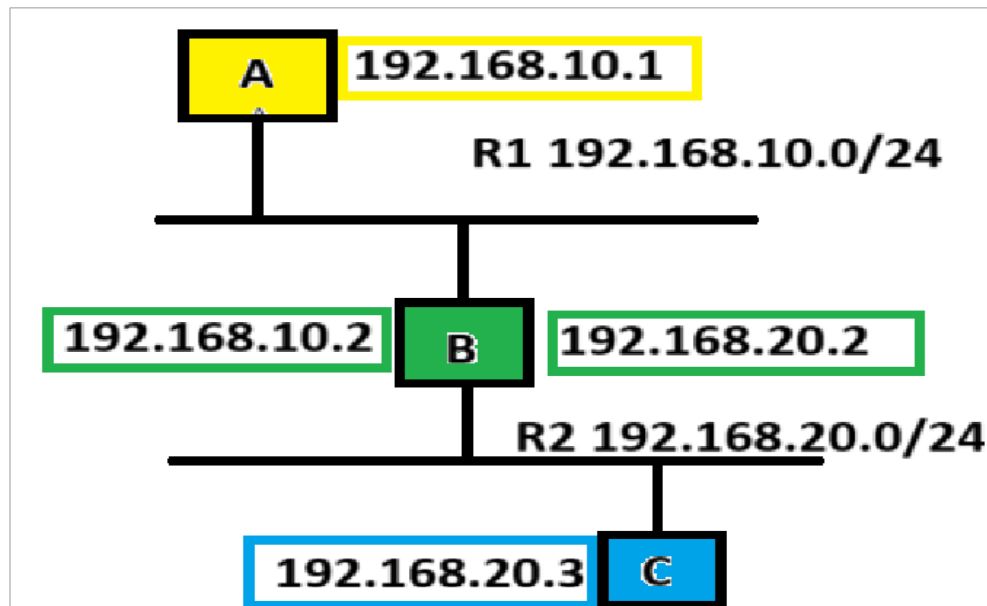
- **Réseau R1 (192.168.10.0/24)** : configuré en mode **NAT**
- **Réseau R2 (192.168.20.0/24)** : configuré en mode **Host-Only**

Les machines virtuelles sont réparties comme suit :

- **Machine A** : une seule carte réseau connectée à R1, avec l'adresse IP **192.168.10.1**.
- **Machine B** : deux cartes réseau, la première sur R1 avec l'adresse **192.168.10.2**, et la seconde sur R2 avec l'adresse **192.168.20.2**. Cette machine jouera un rôle de **passerelle** entre les deux réseaux.
- **Machine C** : une seule carte réseau connectée à R2, avec l'adresse IP **192.168.20.3**.

1 Configuration des machines A, B et C

1.1 Maquette du réseau :



1.1.1 Machine A

```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:4c:5d:a5 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.10.1/24 brd 192.168.10.255 scope global ens33
        valid_lft forever preferred_lft forever
```

```
root@debian:~# ip route
default via 192.168.10.2 dev ens33 onlink
192.168.10.0/24 dev ens33 proto kernel scope link src 192.168.10.1
```

1.1.2 Machine B

```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:a0:a6:a3 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.10.2/24 brd 192.168.10.255 scope global ens33
        valid_lft forever preferred_lft forever
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:a0:a6:ad brd ff:ff:ff:ff:ff:ff
    altname enp2s2
    inet 192.168.20.2/24 brd 192.168.20.255 scope global ens34
        valid_lft forever preferred_lft forever
```

```
root@debian:~# ip route
192.168.10.0/24 dev ens33 proto kernel scope link src 192.168.10.2
192.168.20.0/24 dev ens34 proto kernel scope link src 192.168.20.2
```

1.1.3 Machine C

```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:7f:4a:a7 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.20.3/24 brd 192.168.20.255 scope global ens33
        valid_lft forever preferred_lft forever
```

```
root@debian:~# ip route
default via 192.168.20.2 dev ens33 onlink
192.168.20.0/24 dev ens33 proto kernel scope link src 192.168.20.3
```

2 Questions Réponses

2.1 Question 1

Comment pouvez-vous savoir, quelle carte réseau linux de B est la carte en mode HostOnly et quelle carte réseau linux de B est en mode NAT ? Ainsi, si les 2 cartes vues par Linux sont appelées ens33 et ens37 (le nom dépend de votre version de vmware). Comment savoir si c'est ens33 qui est en NAT ou si c'est ens37 ?

2.1.1 Réponse 1

L'identification de l'interface réseau assignée à la machine virtuelle B s'effectue par vérification des adresses MAC, en les comparant avec celles renseignées dans la configuration VMWare.

The top screenshot shows the VMware Workstation interface for a virtual machine. The 'Network Adapter' is configured as 'NAT'. The 'MAC Address' field displays '00:0C:29:A0:A6:A3'. The terminal window on the right shows the following network configuration:

```
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
inet6 ::1/128 scope host noprefixroute
    valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc f
link/ether 00:0c:29:a0:a6:a3 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
inet 192.168.10.2/24 brd 192.168.10.255 scope global ens
    valid_lft forever preferred_lft forever
```

The bottom screenshot shows the same interface but with the 'Network Adapter' configured as 'Host-only'. The 'MAC Address' field displays '00:0C:29:A0:A6:AD'. The terminal window on the right shows the following network configuration:

```
inet 192.168.10.2/24 brd 192.168.10.255 scope global ens
    valid_lft forever preferred_lft forever
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc f
link/ether 00:0c:29:a0:a6:ad brd ff:ff:ff:ff:ff:ff
    altname enp2s2
inet 192.168.20.2/24 brd 192.168.20.255 scope global ens
    valid_lft forever preferred_lft forever
root@debian:~#
```

2.2 Question 2

Qu'est ce qu'il faut faire pour que la machine A puisse communiquer avec la machine C ?

2.2.1 Réponse 2

Sur la machine A, après avoir configuré son adresse on lui donne comme adresse de passerelle l'adresse de la machine B/R1. La même chose pour la machine C avec l'adresse de B/R2, puis d'activer le routage des paquets sur la machine B. L'activation du forwarding permet de la faire opérer en tant que routeur permettant ainsi aux machines A et C de communiquer.

Ping Machine A vers B/R1, B/R2 et C:

```
root@debian:~# ping -c 4 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.539 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.713 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=0.765 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.825 ms

--- 192.168.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3051ms
rtt min/avg/max/mdev = 0.539/0.710/0.825/0.106 ms
root@debian:~# ping -c 4 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=64 time=0.604 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=64 time=0.667 ms
64 bytes from 192.168.20.2: icmp_seq=3 ttl=64 time=0.488 ms
64 bytes from 192.168.20.2: icmp_seq=4 ttl=64 time=0.682 ms

--- 192.168.20.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3062ms
rtt min/avg/max/mdev = 0.488/0.610/0.682/0.076 ms
root@debian:~# ping -c 4 192.168.20.3
PING 192.168.20.3 (192.168.20.3) 56(84) bytes of data.
64 bytes from 192.168.20.3: icmp_seq=1 ttl=63 time=1.22 ms
64 bytes from 192.168.20.3: icmp_seq=2 ttl=63 time=1.45 ms
64 bytes from 192.168.20.3: icmp_seq=3 ttl=63 time=1.29 ms
64 bytes from 192.168.20.3: icmp_seq=4 ttl=63 time=1.38 ms

--- 192.168.20.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.216/1.334/1.453/0.090 ms
root@debian:~#
```

Ping Machine B vers B/R1, B/R2

```
root@debian:~# ping -c 4 192.168.20.2
PING 192.168.20.2 (192.168.20.2) 56(84) bytes of data.
64 bytes from 192.168.20.2: icmp_seq=1 ttl=64 time=0.574 ms
64 bytes from 192.168.20.2: icmp_seq=2 ttl=64 time=0.655 ms
64 bytes from 192.168.20.2: icmp_seq=3 ttl=64 time=0.723 ms
64 bytes from 192.168.20.2: icmp_seq=4 ttl=64 time=0.749 ms

--- 192.168.20.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3060ms
rtt min/avg/max/mdev = 0.574/0.675/0.749/0.067 ms
root@debian:~# ping -c 4 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.620 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.834 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=0.801 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.737 ms

--- 192.168.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3055ms
rtt min/avg/max/mdev = 0.620/0.748/0.834/0.081 ms
root@debian:~# █
```

2.3 Question 3

Est-il intéressant de vérifier aussi que depuis C, un ping vers 192.168.10.1 (A) est OK ?

2.3.1 Réponse 3

Il n'est pas nécessaire d'effectuer un test de ping de la machine C vers la machine A, la communication bidirectionnelle étant déjà démontrée. En effet, la commande **ping** utilise le protocole **ICMP** pour établir sa communication. Lorsque la machine A exécute un ping vers la machine C, elle envoie un paquet **ICMP Echo Request**. Une réponse sous la forme d'un paquet **ICMP Echo Reply** de la machine C atteste non seulement de la connectivité du chemin A vers C, mais aussi de la capacité de C à répondre à A, validant ainsi la connectivité retour. Par conséquent, le succès du **ping** de A vers C confirme que la couche réseau entre les deux hôtes est fonctionnelle dans les deux sens.

2.4 Question 4

on vous demande d'expliquer ce que permet de vérifier chacune des 3 étapes décrites ci-dessus. On part du principe qu'une étape n'est testée que si les précédentes sont OK.

2.4.1 Réponse 4

Chaque étape permet de savoir si tous les machines communiquent entre elles et de vérifier que la machine B se comporte bien comme un routeur entre la machine A et C.

3 Trajet d'un paquet « ping » de A à C : IP, MAC, TTL

3.1.1 Analyse du Processus de Routage et des En-têtes de Paquet

3.1.2 Décision de Routage et Transfert

- **Au niveau de la Machine A :** La machine A détermine que l'adresse IP de destination (machine C) ne se trouve pas sur son réseau local. Conformément à sa table de routage, elle route donc le paquet vers sa **passerelle par défaut** (l'interface R1 de la machine B).
- **Au niveau de la Machine B (agissant comme routeur) :** La machine B reçoit le paquet sur son interface R1. Elle consulte sa propre table de routage et constate que le réseau de destination (celui de la machine C) est directement connecté à son interface R2. Elle **transfère** donc le paquet vers cette interface pour qu'il soit délivré à la machine C.

3.1.3 Identification des Paquets

L'identification unique d'un paquet spécifique à travers les routeurs (R1 et R2) est garantie principalement par une combinaison de champs dans l'en-tête IP et TCP, notamment :

- **Les adresses IP source et destination.**
- **Le numéro de séquence** (pour les protocoles de transport comme TCP).
- **L'identifiant de fragment** (pour les paquets IP).

Cette combinaison permet de suivre et de réassembler le flux de données de manière fiable.

3.1.4 Time to live (TTL)

Le **TTL** (Time To Live) est un champ de l'en-tête IP qui représente le nombre maximal de sauts (hops) autorisés pour un paquet à travers le réseau. À chaque passage par un routeur (que ce soit R1, R2 ou tout autre), cette valeur est décrétementée de 1. Si elle atteint zéro, le paquet est supprimé et un message ICMP "**Time Exceeded**" est renvoyé à l'expéditeur. Ce mécanisme empêche les paquets de circuler indéfiniment.

3.1.5 Modification des En-têtes lors du Routage

Les adresses IP de niveau 3 restent constantes pendant tout le trajet du paquet. Seules les adresses MAC de niveau 2 changent à chaque saut, car elles identifient toujours le prochain saut immédiat sur le segment de réseau local :

- **Chemin A → C :**

- ▶ **Sur le segment R1 (A → B) :**

- **MAC Source :** Adresse MAC de la machine A
- **MAC Destination :** Adresse MAC de l'interface R1 de la machine B
- **IP Source :** Adresse IP de A (constante)
- **IP Destination :** Adresse IP de C (constante)

- ▶ **Sur le segment R2 (B → C) :**

- **MAC Source :** Adresse MAC de l'interface R2 de la machine B
- **MAC Destination :** Adresse MAC de la machine C
- **IP Source :** Adresse IP de A (constante)
- **IP Destination :** Adresse IP de C (constante)

- **Chemin de retour C → A :**

- ▶ **Sur le segment R2 (C → B) :**

- **MAC Source :** Adresse MAC de la machine C
- **MAC Destination :** Adresse MAC de l'interface R2 de la machine B
- **IP Source :** Adresse IP de C (constante)
- **IP Destination :** Adresse IP de A (constante)

- ▶ **Sur le segment R1 (B → A) :**

- **MAC Source :** Adresse MAC de l'interface R1 de la machine B
- **MAC Destination :** Adresse MAC de la machine A
- **IP Source :** Adresse IP de C (constante)
- **IP Destination :** Adresse IP de A (constante)

3.2 Les Paquets

3.2.1 Ping de A vers C sur R1

Time	Source	Destination	Protocol	Length	Info
1 0.000000000	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=1/256, ttl=64
2 0.000664902	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=1/256, ttl=63
3 1.002059856	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=2/512, ttl=64
4 1.002678389	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=2/512, ttl=63
5 2.004184303	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=3/768, ttl=64
6 2.004832012	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=3/768, ttl=63
7 3.006455279	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=4/1024, ttl=6
8 3.006883902	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=4/1024, ttl=6

<ul style="list-style-type: none"> Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface Ethernet II, Src: VMware_4c:5d:a5 (00:0c:29:4c:5d:a5), Dst: VMware_a0:a6:a3 (00:0c:29:a0:a6:a3) <ul style="list-style-type: none"> Destination: VMware_a0:a6:a3 (00:0c:29:a0:a6:a3) <i>B/R1</i> Source: VMware_4c:5d:a5 (00:0c:29:4c:5d:a5) <i>A</i> Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.20.3 Internet Control Message Protocol 	<pre> 0000 00 0c 29 a0 a6 a3 0010 00 54 35 65 40 00 0020 14 03 08 00 19 68 0030 00 00 04 b2 00 00 0040 16 17 18 19 1a 1b 0050 26 27 28 29 2a 2b 0060 36 37 </pre>
---	--

3.2.2 Ping de A vers C sur R2

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=1/256, ttl=63 (reply in 2)
2	0.000627109	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=1/256, ttl=64 (request in 1)
3	1.002056058	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=2/512, ttl=63 (reply in 4)
4	1.002643352	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=2/512, ttl=64 (request in 3)
5	2.004199281	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=3/768, ttl=63 (reply in 6)
6	2.004796925	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=3/768, ttl=64 (request in 5)
7	3.006450559	192.168.10.1	192.168.20.3	ICMP	98	Echo (ping) request id=0x0b8c, seq=4/1024, ttl=63 (reply in 8)
8	3.006853323	192.168.20.3	192.168.10.1	ICMP	98	Echo (ping) reply id=0x0b8c, seq=4/1024, ttl=64 (request in 7)

<ul style="list-style-type: none"> Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface ens34, Ethernet II, Src: VMware_a0:a6:ad (00:0c:29:a0:a6:ad), Dst: VMware_7f:4a:a7 (00:0c:29:7f:4a:a7) <ul style="list-style-type: none"> Destination: VMware_7f:4a:a7 (00:0c:29:7f:4a:a7) <i>B/R2</i> Source: VMware_a0:a6:ad (00:0c:29:a0:a6:ad) <i>C</i> Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.20.3 Internet Control Message Protocol 	<pre> 0000 00 0c 29 7f 4a a7 00 0c 29 a0 a6 ad 00 0010 00 54 35 65 40 00 3f 01 66 ef c0 a8 00 0020 14 03 08 00 19 68 0b 8c 00 01 4b 1d c0 0030 00 00 04 b2 00 00 00 00 00 00 10 11 12 0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 0060 36 37 </pre>
--	--

3.2.3 SSH de A vers C sur R1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.10.1	192.168.20.3	TCP	74	54240 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=14
2	0.003554814	192.168.20.3	192.168.10.1	TCP	74	22 → 54240 [SYN, ACK] Seq=0 Ack=1 Win=65160 L
3	0.004178761	192.168.10.1	192.168.20.3	TCP	66	54240 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0
4	0.006159566	192.168.10.1	192.168.20.3	SSHv2	106	Client: Protocol (SSH-2.0-OpenSSH 9.2p1 Debia
5	0.008810190	192.168.20.3	192.168.10.1	TCP	66	22 → 54240 [ACK] Seq=1 Ack=41 Win=65152 Len=0
6	0.128742692	192.168.20.3	192.168.10.1	SSHv2	106	Server: Protocol (SSH-2.0-OpenSSH 9.2p1 Debia
7	0.129235099	192.168.10.1	192.168.20.3	TCP	66	54240 → 22 [ACK] Seq=41 Ack=41 Win=64256 Len=
8	0.130286534	192.168.10.1	192.168.20.3	SSHv2	16...	Client: Key Exchange Init
9	0.131483181	192.168.20.3	192.168.10.1	TCP	66	22 → 54240 [ACK] Seq=41 Ack=1601 Win=64000 Le

<ul style="list-style-type: none"> Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface en Ethernet II, Src: VMware 4c:5d:a5 (00:0c:29:4c:5d:a5), Dst: VMware_a0:a6:a3 (00:0c:29:a0:a6:a3) <ul style="list-style-type: none"> Destination: VMware_a0:a6:a3 (00:0c:29:a0:a6:a3) Source: VMware_4c:5d:a5 (00:0c:29:4c:5d:a5) Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.20.3 Transmission Control Protocol, Src Port: 54240, Dst Port: 22, Seq: 1, Ack: 1, Len: <ul style="list-style-type: none"> Source Port: 54240 Destination Port: 22 [Stream index: 0] [Conversation completeness: Incomplete, DATA (15)] [TCP Segment Len: 0] Sequence Number: 1 (relative sequence number) Sequence Number (raw): 1528443203 [Next Sequence Number: 1 (relative sequence number)] Acknowledgment Number: 1 (relative ack number) Acknowledgment number (raw): 1587059720 1000 = Header Length: 32 bytes (8) Flags: 0x010 (ACK) 	<pre> 0000 00 0c 29 a0 a6 0010 00 34 2c a0 40 0020 14 03 d3 e0 00 0030 01 f6 68 07 00 0040 92 bc </pre>
---	--

3.2.4 SSH de A vers C sur R2

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.10.1	192.168.20.3	TCP	74	54240 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=14
2	0.003493348	192.168.20.3	192.168.10.1	TCP	74	22 → 54240 [SYN, ACK] Seq=0 Ack=1 Win=65160 L
3	0.004142543	192.168.10.1	192.168.20.3	TCP	66	54240 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0
4	0.006127185	192.168.10.1	192.168.20.3	SSHv2	106	Client: Protocol (SSH-2.0-OpenSSH 9.2p1 Debia
5	0.008722133	192.168.20.3	192.168.10.1	TCP	66	22 → 54240 [ACK] Seq=1 Ack=41 Win=65152 Len=0

<ul style="list-style-type: none"> Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface en Ethernet II, Src: VMware_a0:a6:ad (00:0c:29:a0:a6:ad), Dst: VMware_7f:4a:a7 (00:0c:29:7f:4a:a7) <ul style="list-style-type: none"> Destination: VMware_7f:4a:a7 (00:0c:29:7f:4a:a7) Source: VMware_a0:a6:ad (00:0c:29:a0:a6:ad) Type: IPv4 (0x0800) Internet Protocol Version 4, Src: 192.168.10.1, Dst: 192.168.20.3 Transmission Control Protocol, Src Port: 54240, Dst Port: 22, Seq: 1, Ack: 1, Len: <ul style="list-style-type: none"> Source Port: 54240 Destination Port: 22 [Stream index: 0] [Conversation completeness: Incomplete, DATA (15)] [TCP Segment Len: 0] Sequence Number: 1 (relative sequence number) Sequence Number (raw): 1528443203 [Next Sequence Number: 1 (relative sequence number)] Acknowledgment Number: 1 (relative ack number) Acknowledgment number (raw): 1587059720 1000 = Header Length: 32 bytes (8) Flags: 0x010 (ACK) Window: 502 	<pre> 0000 00 0c 29 7f 4a 0010 00 34 2c a0 40 0020 14 03 d3 e0 00 0030 01 f6 68 07 00 0040 92 bc </pre>
--	--

La modification des en-têtes lors du routage des paquets de la machine A vers la machine C en utilisant le **ssh** est exactement la même que celui du routage des paquet de A vers C avec la commande **ping**. Et l'identification des paquets sur R1 et R2 se fait à l'aide du **numéro de séquence** permettant d'ordonner les segments reçus et Le **numéro d'acquittement** (ACK) confirme la bonne réception des données.

Documentation :

https://docs.google.com/document/d/1Sf9spCkk1uyEMgLwvrEgMITHBo7yCXIqV_upaiNqlwE/edit?tab=t.0#heading=h.ns88yq53dleq

Merci pour la lecture