

Université de Toulon

IUT de Toulon

Département Génie Electrique et Informatique Industrielle (GEII)

Réseaux Cybersécurité Unbuntu

Rapport de TP

écrit le 5 decembre 2023

par

Bruno HANNA

Encadrant universitaire : Stephane PIGNOL



Table des matières

Introduction	1
Chapitre 1 Partie configuration Unbutu	2
1.1 Pour Démarrer	2
1.1.1 Identification des Interfaces Actives et Configuration IP	2
1.1.2 Détermination de la Passerelle par Défaut	2
1.1.3 Établissement d'une Connexion Filaire en IP Fixe	3
1.1.4 Transition vers une Configuration en IP Dynamique	3
1.1.5 Résolution de Noms de Domaine via DNS	3
1.1.6 Recherche de Nom de Domaine par Adresse IP	4
1.1.7 Détermination de l'Adresse IP du Serveur Google.com	4
1.1.8 Utilisation de la Commande Dig pour Interroger les Serveurs DNS	6
1.1.9 Optimisation de la Lisibilité des Résultats DNS avec Dig	6
1.1.10 Test de Connectivité avec la Commande Ping	7
1.1.11 Variation de la Taille des Paquets ICMP avec la Commande Ping	7
1.1.12 Utilisation de l'Option -M do avec la Commande Ping	8
1.1.13 Détermination de la Taille Maximale de Données sans Fragmentation	9
1.1.14 Table des Entrées ARP	10
Conclusion	12

Introduction

L'intégrité et la performance des réseaux informatiques sont essentielles dans le monde interconnecté d'aujourd'hui. Ce compte rendu examine divers aspects de la gestion et de la vérification de la connectivité réseau, en se concentrant sur des commandes et des procédures essentielles. Nous abordons des sujets tels que la résolution ARP, la vérification de la connectivité avec ping, la manipulation de la taille des paquets et l'utilisation de l'option -M do. Nous explorons également l'importance de la table des entrées ARP dans la communication réseau. L'objectif de ce compte rendu est de fournir un aperçu complet des outils et des concepts nécessaires pour maintenir et optimiser un réseau informatique.

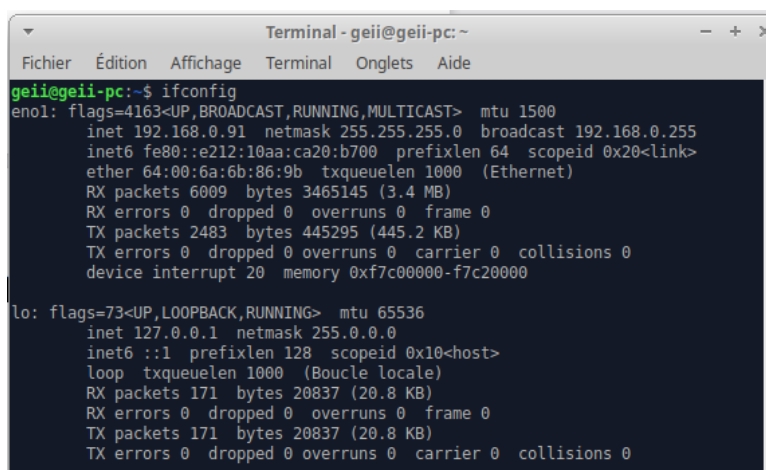
Partie configuration Unbutu

1.1 Pour Démarrer

1.1.1 Identification des Interfaces Actives et Configuration IP

L'ouverture d'un terminal s'effectue par la combinaison des touches CTRL + ALT + t. Cette action est préliminaire à l'exécution de la commande `ifconfig`, permettant l'identification des interfaces réseau actives. Typiquement, l'interface de boucle locale est désignée par `lo` tandis que l'interface filaire est souvent nommée `enp3s0` ou similaire, suivant la nomenclature système.

La commande `ifconfig` révèle les détails de configuration réseau de l'interface filaire, incluant l'adresse IP, le masque de sous-réseau (`netmask`), et l'adresse de broadcast. Ces informations sont critiques pour le diagnostic et la configuration réseau.



```
Terminal - geii@geii-pc: ~
Fichier  Édition  Affichage  Terminal  Onglets  Aide

geii@geii-pc:~$ ifconfig
eno1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 192.168.0.91 netmask 255.255.255.0  broadcast 192.168.0.255
    inet6 fe80::e212:10aa:ca20:b700 prefixlen 64 scopeid 0x20<link>
    ether 64:00:6a:6b:86:9b txqueuelen 1000 (Ethernet)
    RX packets 6009  bytes 3465145 (3.4 MB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 2483  bytes 445295 (445.2 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
    device interrupt 20  memory 0xf7c00000-f7c20000

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 (Boucle locale)
    RX packets 171  bytes 20837 (20.8 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 171  bytes 20837 (20.8 KB)
    TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Figure 1.1 – ifconfig

1.1.2 Détermination de la Passerelle par Défaut

La passerelle par défaut est l'adresse IP par laquelle le système envoie des paquets vers des réseaux extérieurs. Pour identifier cette adresse cruciale pour le routage, la commande `route -n` est employée dans le terminal. La colonne 'Gateway' de la sortie affiche l'adresse de la passerelle par défaut.

Dans le cas présent, la passerelle par défaut identifiée est 192.168.0.1. Cette adresse est généralement attribuée au routeur ou au dispositif de couche 3 qui gère le trafic sortant vers d'autres sous-réseaux ou vers l'Internet.

```
geii@geii-pc:~$ netstat -rn
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic  MSS Fenêtre  irtt  Iface
0.0.0.0          192.168.0.1     0.0.0.0          UG     0 0        0 enol
169.254.0.0      0.0.0.0         255.255.0.0      U      0 0        0 enol
192.168.0.0      0.0.0.0         255.255.255.0    U      0 0        0 enol
geii@geii-pc:~$ netstat -nr
Table de routage IP du noyau
Destination      Passerelle      Genmask          Indic  MSS Fenêtre  irtt  Iface
0.0.0.0          192.168.0.1     0.0.0.0          UG     0 0        0 enol
169.254.0.0      0.0.0.0         255.255.0.0      U      0 0        0 enol
192.168.0.0      0.0.0.0         255.255.255.0    U      0 0        0 enol
geii@geii-pc:~$ ip route show
default via 192.168.0.1 dev enol proto dhcp metric 100
169.254.0.0/16 dev enol scope link metric 1000
192.168.0.0/24 dev enol proto kernel scope link src 192.168.0.91 metric 100
```

Figure 1.2 – Capture d'écran de la commande `route -n` montrant la passerelle par défaut.

1.1.3 Établissement d'une Connexion Filaire en IP Fixe

La configuration d'une adresse IP fixe est réalisée à travers l'interface graphique `networkmanager`. Un clic gauche sur l'icône de réseau dans la barre d'état ouvre le menu de gestion des connexions. L'option `nm-connection-editor` est également disponible pour accéder directement à l'éditeur de connexions.

La création d'une nouvelle connexion filaire statique est effectuée en sélectionnant le symbole `+`, puis en ajustant les paramètres réseau selon les spécifications du tableau de l'Annexe A. La commande `ifconfig` permet de vérifier l'adresse IP attribuée et de confirmer la configuration réseau.

1.1.4 Transition vers une Configuration en IP Dynamique

La gestion d'une adresse IP dynamique s'effectue par la modification de la connexion existante en une Connexion Ethernet `dhcp` via l'interface graphique ou l'éditeur de connexions `networkmanager`. Ceci est essentiel pour permettre au système d'obtenir automatiquement une adresse IP du serveur DHCP.

La commande `sudo dhclient -r` est utilisée pour libérer l'adresse IP actuelle et réinitialiser la configuration réseau. Par la suite, l'exécution de `sudo dhclient` permet de demander une nouvelle adresse IP. Cela garantit que les paramètres réseau sont actualisés et correctement assignés par le serveur DHCP.

1.1.5 Résolution de Noms de Domaine via DNS

Le système de noms de domaine (DNS) est un service essentiel permettant d'associer les noms de domaine à leurs adresses IP correspondantes. Le programme `nslookup`, conforme aux RFC 1034, 1035 et 1033, est utilisé pour interroger les serveurs DNS.

Pour identifier l'adresse IP associée au nom de domaine `facebook.com`, la commande `nslookup facebook.com` est exécutée. Le résultat affiche l'adresse IP attribuée au nom de domaine spécifié, facilitant ainsi la connexion réseau vers le domaine ciblé.

```
geii@geii-pc:~$ ifconfig
enol: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.91 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a394:89a2:9d6b:8568 prefixlen 64 scopeid 0x20<link>
    ether 64:00:6a:6b:86:9b txqueuelen 1000 (Ethernet)
    RX packets 24315 bytes 7428521 (7.4 MB)
    RX errors 0 dropped 7 overruns 0 frame 0
    TX packets 5003 bytes 786611 (786.6 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7c00000-f7c20000

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 (Boucle locale)
    RX packets 1589 bytes 138570 (138.5 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1589 bytes 138570 (138.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

geii@geii-pc:~$ ifconfig
enol: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.205 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::5232:dfd5:b686:a906 prefixlen 64 scopeid 0x20<link>
    ether 64:00:6a:6b:86:9b txqueuelen 1000 (Ethernet)
    RX packets 24425 bytes 7447798 (7.4 MB)
    RX errors 0 dropped 7 overruns 0 frame 0
    TX packets 5134 bytes 807031 (807.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7c00000-f7c20000

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 (Boucle locale)
    RX packets 1703 bytes 152178 (152.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1703 bytes 152178 (152.1 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 1.3 – Confirmation de la configuration IP fixe via ifconfig.

1.1.6 Recherche de Nom de Domaine par Adresse IP

La commande `nslookup` est également utilisée pour effectuer une recherche à l'inverse, c'est-à-dire trouver le nom de domaine associé à une adresse IP donnée. Cela est particulièrement utile pour vérifier les informations de nommage associées aux adresses IP.

En exécutant la commande `nslookup 137.129.43.129`, il est possible de déterminer le nom de la machine correspondant à cette adresse IP spécifique. Les résultats indiquent le nom de domaine complet et fournissent également des informations sur le serveur de noms autoritatif pour ce domaine.

1.1.7 Détermination de l'Adresse IP du Serveur Google.com

La commande `nslookup` permet de résoudre les adresses IP associées aux noms de domaine. Pour le domaine `google.com`, l'exécution de cette commande fournit l'adresse IP du serveur qui héberge le domaine. Cela confirme la possibilité d'établir une connexion réseau avec le serveur de

```
geii@geii-pc:~$ sudo dhclient -r
Killed old client process
geii@geii-pc:~$ ifconfig
enol: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::a394:89a2:9d6b:8568 prefixlen 64 scopeid 0x20<link>
    ether 64:00:6a:6b:86:9b txqueuelen 1000 (Ethernet)
    RX packets 26101 bytes 7649367 (7.6 MB)
    RX errors 0 dropped 7 overruns 0 frame 0
    TX packets 5389 bytes 845558 (845.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 20 memory 0xf7c00000-f7c20000

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 (Boucle locale)
    RX packets 2225 bytes 193898 (193.8 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2225 bytes 193898 (193.8 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 1.4 – Résultat de la commande `sudo dhclient -r` suivie de `sudo dhclient` pour réinitialiser et renouveler l'adresse IP.

```
geii@geii-pc:~$ nslookup facebook.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   facebook.com
Address: 185.60.219.35
Name:   facebook.com
Address: 2a03:2880:f17b:88:face:b00c:0:25de
```

Figure 1.5 – Résultat de la commande `nslookup` pour le domaine `facebook.com`.

```
geii@geii-pc:~$ nslookup 137.129.43.129
129.43.129.137.in-addr.arpa    name = weather.gmdss.org.
129.43.129.137.in-addr.arpa    name = archivesduclimat.meteo-france.fr.
129.43.129.137.in-addr.arpa    name = archivesduclimat.meteofrance.fr.
129.43.129.137.in-addr.arpa    name = cyclones.meteo.re.
129.43.129.137.in-addr.arpa    name = www.meteo.fr.
129.43.129.137.in-addr.arpa    name = archivesduclimat.meteo-france.eu.
129.43.129.137.in-addr.arpa    name = www.gmdss.org.
129.43.129.137.in-addr.arpa    name = cyclone.meteo.re.
129.43.129.137.in-addr.arpa    name = archivesduclimat.meteofrance.eu.

Authoritative answers can be found from:
129.137.in-addr.arpa    nameserver = nscf2.meteo.fr.
129.137.in-addr.arpa    nameserver = nscf1.meteo.fr.
nscf2.meteo.fr    internet address = 162.159.27.108
nscf2.meteo.fr    has AAAA address 2400:cb00:2049:1::a29f:1b6c
nscf1.meteo.fr    internet address = 162.159.26.51
nscf1.meteo.fr    has AAAA address 2400:cb00:2049:1::a29f:1a33
```

Figure 1.6 – Résultat de la commande `nslookup` pour l'adresse IP `137.129.43.129`.

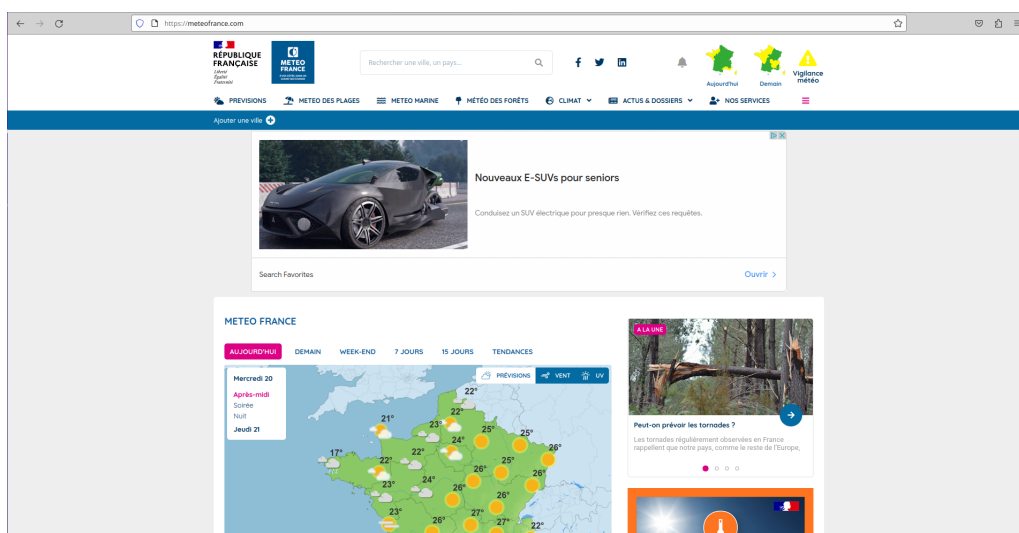


Figure 1.7 – Résultat de la commande nslookup pour l'adresse IP 137.129.43.129.

```
geii@geii-pc:~$ nslookup google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.211.206
Name:   google.com
Address: 2a00:1450:4006:809::200e
```

Figure 1.8 – Résultat de la commande nslookup pour le domaine google.com.

1.1.8 Utilisation de la Commande Dig pour Interroger les Serveurs DNS

La commande dig est un outil flexible pour interroger les serveurs DNS. Elle fournit des informations détaillées sur le domaine spécifié, tel que google.com. Cet outil est utilisé pour obtenir des réponses autoritatives concernant les adresses IP associées à un nom de domaine, ainsi que les serveurs de noms responsables de ce domaine.

Pour google.com, la commande dig google.com retourne non seulement l'adresse IP du serveur correspondant, mais aussi une liste de tous les serveurs de noms (NS) associés, ainsi que des enregistrements supplémentaires tels que les adresses IPv6 (AAAA).

1.1.9 Optimisation de la Lisibilité des Résultats DNS avec Dig

La commande dig est utilisée pour interroger les serveurs DNS. L'ajout de l'option +short à la fin de la commande permet d'obtenir une réponse concise, affichant uniquement l'adresse IP associée à un nom d'hôte donné. Cette pratique est recommandée pour épurée les résultats, facilitant ainsi l'analyse rapide des enregistrements DNS.


```

geii@geii-pc:~$ dig google.com

; <<> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<> google.com
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 59012
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                78      IN      A      216.58.211.206

;; AUTHORITY SECTION:
google.com.                78      IN      NS      ns3.google.com.
google.com.                78      IN      NS      ns4.google.com.
google.com.                78      IN      NS      ns2.google.com.
google.com.                78      IN      NS      ns1.google.com.

;; ADDITIONAL SECTION:
ns2.google.com.            78      IN      A      216.239.34.10
ns1.google.com.            78      IN      A      216.239.32.10
ns1.google.com.            78      IN      AAAA   2001:4860:4802:32::a
ns3.google.com.            78      IN      AAAA   2001:4860:4802:36::a
ns4.google.com.            78      IN      A      216.239.38.10
ns3.google.com.            78      IN      A      216.239.36.10
ns4.google.com.            78      IN      AAAA   2001:4860:4802:38::a
ns2.google.com.            78      IN      AAAA   2001:4860:4802:34::a

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Wed Sep 20 17:04:44 CEST 2023
;; MSG SIZE rcvd: 303

```

Figure 1.9 – Résultat de la commande dig pour le domaine google.com.

1.1.10 Test de Connectivité avec la Commande Ping

Le test de connectivité est un élément fondamental de la vérification du réseau. La commande ping permet de vérifier la disponibilité d'une machine sur le réseau en envoyant des paquets ICMP (Internet Control Message Protocol). La réponse positive de la commande ping, indiquant le temps aller-retour en millisecondes, confirme la bonne communication entre les hôtes.

1.1.11 Variation de la Taille des Paquets ICMP avec la Commande Ping

La commande ping, outre son rôle dans la vérification de la connectivité, offre la possibilité de spécifier la taille des paquets envoyés à l'aide de l'option `-s`. Cela permet d'évaluer la performance du réseau avec des charges de différentes tailles. Les tests sont réalisés en spécifiant successivement 100, 1000 et 2000 octets de données, permettant ainsi de mesurer l'influence de la taille du paquet sur le temps de réponse.

```

geii@geii-pc:~$ dig -x +short 138.129.43.129
; <<>> DiG 9.18.12-0ubuntu0.22.04.2-Ubuntu <<>> -x +short 138.129.43.129
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 808
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;+short.in-addr.arpa.          IN      PTR

;; AUTHORITY SECTION:
in-addr.arpa.                3600    IN      SOA      b.in-addr-servers.arpa. nstld.iana.org. 2022092189 1800 900 604800 3600

;; Query time: 31 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Wed Sep 20 17:09:10 CEST 2023
;; MSG SIZE rcvd: 116

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 35967
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;138.129.43.129.              IN      A

;; AUTHORITY SECTION:
.                            10800   IN      SOA      a.root-servers.net. nstld.verisign-grs.com. 2023092000 1800 900 604800 86400

;; Query time: 16 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Wed Sep 20 17:09:10 CEST 2023
;; MSG SIZE rcvd: 118
geii@geii-pc:~$ dig +short google.com
216.58.211.206

```

Figure 1.10 – Exemple de sortie de la commande dig avec l'option +short.

```

geii@geii-pc:~$ ping 192.168.0.70
PING 192.168.0.70 (192.168.0.70) 56(84) bytes of data.
64 bytes from 192.168.0.70: icmp_seq=1 ttl=64 time=1.65 ms
64 bytes from 192.168.0.70: icmp_seq=2 ttl=64 time=0.857 ms
64 bytes from 192.168.0.70: icmp_seq=3 ttl=64 time=0.780 ms
64 bytes from 192.168.0.70: icmp_seq=4 ttl=64 time=0.816 ms
64 bytes from 192.168.0.70: icmp_seq=5 ttl=64 time=0.839 ms
64 bytes from 192.168.0.70: icmp_seq=6 ttl=64 time=0.789 ms
^C
--- 192.168.0.70 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5085ms
rtt min/avg/max/mdev = 0.780/0.954/1.646/0.310 ms

```

Figure 1.11 – Résultats de la commande ping démontrant la connectivité avec l'hôte cible.

1.1.12 Utilisation de l'Option -M do avec la Commande Ping

L'option -M do de la commande ping permet d'activer la découverte de la fragmentation de paquets interdite. En d'autres termes, elle vérifie si les paquets ICMP de grande taille sont fragmentés lors de leur envoi. Cette option est utile pour évaluer la prise en charge de la fragmentation par les hôtes du réseau.

```
geii@geii-pc:~$ ping 192.168.0.70 -s 100
PING 192.168.0.70 (192.168.0.70) 100(128) bytes of data.
108 bytes from 192.168.0.70: icmp_seq=1 ttl=64 time=0.742 ms
108 bytes from 192.168.0.70: icmp_seq=2 ttl=64 time=0.765 ms
108 bytes from 192.168.0.70: icmp_seq=3 ttl=64 time=0.838 ms
^C
--- 192.168.0.70 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
rtt min/avg/max/mdev = 0.742/0.781/0.838/0.040 ms
geii@geii-pc:~$ ping 192.168.0.70 -s 1000
PING 192.168.0.70 (192.168.0.70) 1000(1028) bytes of data.
1008 bytes from 192.168.0.70: icmp_seq=1 ttl=64 time=1.30 ms
1008 bytes from 192.168.0.70: icmp_seq=2 ttl=64 time=1.20 ms
1008 bytes from 192.168.0.70: icmp_seq=3 ttl=64 time=1.21 ms
^C
--- 192.168.0.70 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.202/1.237/1.298/0.042 ms
geii@geii-pc:~$ ping 192.168.0.70 -s 2000
PING 192.168.0.70 (192.168.0.70) 2000(2028) bytes of data.
2008 bytes from 192.168.0.70: icmp_seq=1 ttl=64 time=1.47 ms
2008 bytes from 192.168.0.70: icmp_seq=2 ttl=64 time=1.44 ms
2008 bytes from 192.168.0.70: icmp_seq=3 ttl=64 time=1.47 ms
^C
--- 192.168.0.70 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.438/1.458/1.472/0.014 ms
```

Figure 1.12 – Résultats de la commande ping pour différentes tailles de paquets.

1.1.13 Détermination de la Taille Maximale de Données sans Fragmentation

Pour déterminer la taille maximale de données pouvant être envoyée sans fragmentation sur une machine du réseau, nous utilisons une approche de dichotomie. En connaissant certaines informations, notamment la taille de l'en-tête ICMP (8 octets) et l'en-tête IP (20 octets), ainsi que la limite MTU (Maximum Transmission Unit) du réseau de 1500 octets, nous pouvons procéder comme suit :

1. Initialement, nous supposons une taille de données de 1500 octets.
2. En déduisant l'en-tête ICMP (8 octets) et l'en-tête IP (20 octets) de cette taille, il reste 1472 octets pour les données.
3. Nous testons la connectivité avec cette taille de données. Si le paquet passe sans fragmentation, alors la taille de 1472 octets est acceptable.

```
geii@geii-pc:~$ ping -M do -s 100 192.168.0.70
PING 192.168.0.70 (192.168.0.70) 100(128) bytes of data.
108 bytes from 192.168.0.70: icmp_seq=1 ttl=64 time=0.839 ms
108 bytes from 192.168.0.70: icmp_seq=2 ttl=64 time=0.680 ms
108 bytes from 192.168.0.70: icmp_seq=3 ttl=64 time=0.859 ms
^C
--- 192.168.0.70 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2028ms
rtt min/avg/max/mdev = 0.680/0.792/0.859/0.080 ms
geii@geii-pc:~$ ping -M do -s 1000 192.168.0.70
PING 192.168.0.70 (192.168.0.70) 1000(1028) bytes of data.
1008 bytes from 192.168.0.70: icmp_seq=1 ttl=64 time=1.22 ms
1008 bytes from 192.168.0.70: icmp_seq=2 ttl=64 time=1.20 ms
1008 bytes from 192.168.0.70: icmp_seq=3 ttl=64 time=1.28 ms
^C
--- 192.168.0.70 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.200/1.231/1.275/0.031 ms
geii@geii-pc:~$ ping -M do -s 2000 192.168.0.70
PING 192.168.0.70 (192.168.0.70) 2000(2028) bytes of data.
ping: local error: message too long, mtu=1500
ping: local error: message too long, mtu=1500
ping: local error: message too long, mtu=1500
^C
--- 192.168.0.70 ping statistics ---
3 packets transmitted, 0 received, +3 errors, 100% packet loss, time 2052ms
```

Figure 1.13 – Exemple de résultats de la commande ping avec l'option -M do.

4. Si le paquet est fragmenté, nous réduisons la taille des données de manière itérative, en diminuant la taille de 100 octets à chaque étape.
5. Nous répétons le test de connectivité jusqu'à ce que nous trouvions la taille maximale sans fragmentation.

La taille maximale de données sans fragmentation déterminée par cette méthode nous permet de justifier la valeur MTU du réseau, qui est limitée à 1500 octets. En effet, en soustrayant les en-têtes ICMP et IP de cette valeur, il reste 1472 octets pour les données, ce qui correspond à la taille maximale sans fragmentation.

1.1.14 Table des Entrées ARP

La table des entrées ARP est un composant essentiel des systèmes informatiques en réseau. Elle joue un rôle crucial dans la résolution d'adresses IP en adresses MAC (Media Access Control). Voici à quoi sert la table des entrées ARP :

La table des entrées ARP, également connue sous le nom de "cache ARP", est une mémoire tampon utilisée pour stocker des associations entre adresses IP et adresses MAC. Elle est utilisée pour résoudre les adresses IP en adresses MAC, ce qui permet aux ordinateurs de communiquer

efficacement sur un réseau local.

Lorsqu'un ordinateur doit communiquer avec un autre ordinateur sur le réseau, il doit connaître l'adresse MAC de la machine de destination pour pouvoir envoyer des données sur le segment local du réseau. La table ARP est utilisée pour maintenir ces associations IP-MAC de manière temporaire. Lorsqu'un ordinateur tente de communiquer avec une adresse IP, il vérifie d'abord la table ARP locale pour voir si l'association IP-MAC nécessaire est déjà présente.

Si l'association IP-MAC n'est pas trouvée dans la table ARP, l'ordinateur utilise un protocole de diffusion ARP pour interroger l'ensemble du réseau et demander à la machine de destination de répondre avec son adresse MAC. Une fois que la réponse est reçue, l'ordinateur met à jour sa table ARP avec cette nouvelle association IP-MAC pour des communications futures.

Conclusion

En conclusion, ce compte rendu a mis en lumière l'importance des connaissances et des procédures liées à la gestion de la connectivité réseau. Nous avons examiné comment résoudre des adresses IP en adresses MAC à l'aide du protocole ARP, comment vérifier la connectivité avec ping, et comment optimiser les tests en ajustant la taille des paquets et en utilisant l'option -M do. De plus, nous avons souligné le rôle crucial de la table des entrées ARP dans la résolution d'adresses IP en adresses MAC.