Mathieu LOSLIER

27 mars 2011

Rapport TP2 WiFi CSMA/CA

Table des matières

Introduction

Dans un réseau local Ethernet classique, la méthode d'accès utilisée par les machines est le CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), pour lequel chaque machine est libre de communiquer à n'importe quel moment. Chaque machine envoyant un message vérifie qu'aucun autre message n'a été envoyé en même temps par une autre machine. Si c'est le cas, les deux machines patientent pendant un temps aléatoire avant de recommencer à émettre.

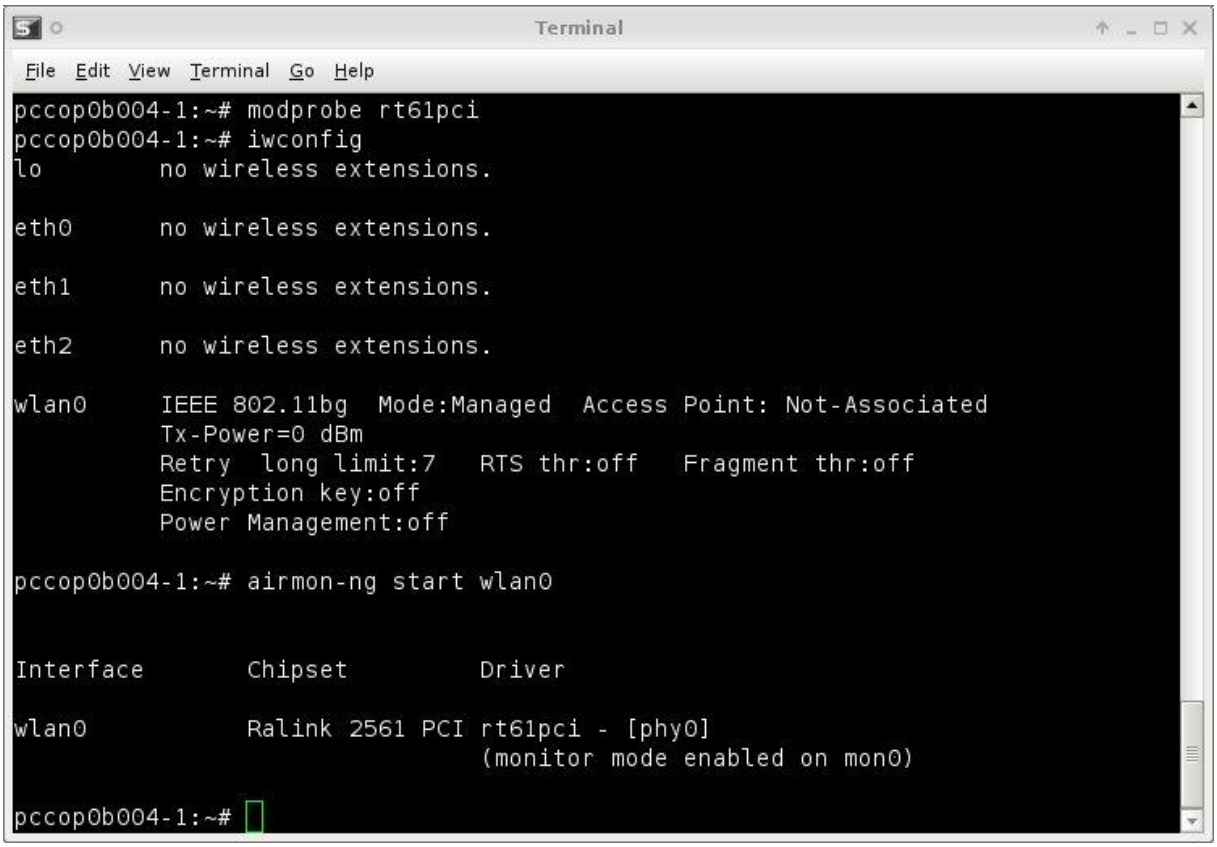
Dans un environnement sans fil ce procédé n'est pas possible dans la mesure où deux stations communiquant avec un récepteur ne s'entendent pas forcément mutuellement en raison de leur rayon de portée. Ainsi la norme 802.11 propose un protocole similaire appelé CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*).

Le protocole CSMA/CA utilise un mécanisme d'esquive de collision basé sur un principe d'accusé de réception réciproque entre l'émetteur et le récepteur.

Ce TP traite de l'étude précise de cette méthode. Pour cela, nous utilisons deux stations en mode ad-hoc, sachant que l'une des stations possède une interface Wifi virtuelle en mode *monitor* afin de sniffer le trafic.

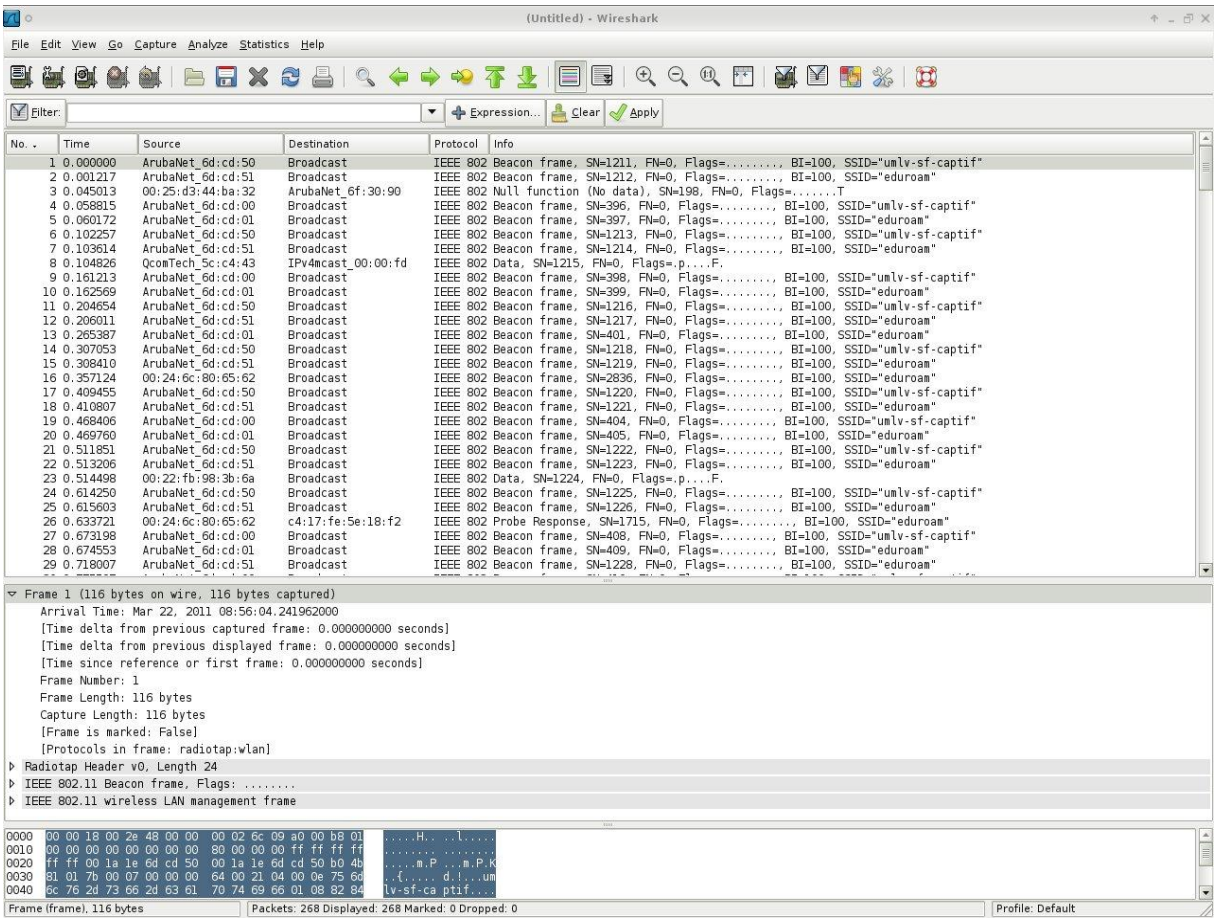
1. Création d'une interface virtuelle

Pour commencer, nous allons créer une interface virtuelle *mon0* en mode monitor pour snifer le trafic**.**



Pour ce faire, il faut charger les modules de la carte réseau (Ralink 2561 dans ce cas) à l'aide de la commande `modprobe rt61pci`. Nous pouvons alors vérifier la présence de l'interface wlan0 qui est active. La création l'interface virtuelle *mon0* s'opère avec la commande `airmon-ng start wlan0`. On peut alors remarquer que l'interface *wlan0* possède une interface virtuelle qui est activée.

On lance une capture wireshark sur l'interface *mon0*.



Nous captons toutes les trames provenant des réseaux WiFi avoisinant.

2. Configuration mode Ad-Hoc

Maintenant nous allons positionner l'interface principale *wlan0* en mode ad-hoc et monter l'interface.

Avant d'effectuer cette manipulation on fait tomber l'interface Ethernet par câble et on débranche le câble.

```
# ifconfig eth0 down
```

On configure la carte pour se connecter en ad-hoc.

```
# wlanconfig wlan0  destroy
# wlanconfig wlan0 create wlandev wifi0 wlanmode adhoc
```

On détruit la configuration actuelle pour se mettre en mode *ad-hoc*. On lui spécifie un nom de réseau et un *channel* (fréquence à utiliser).

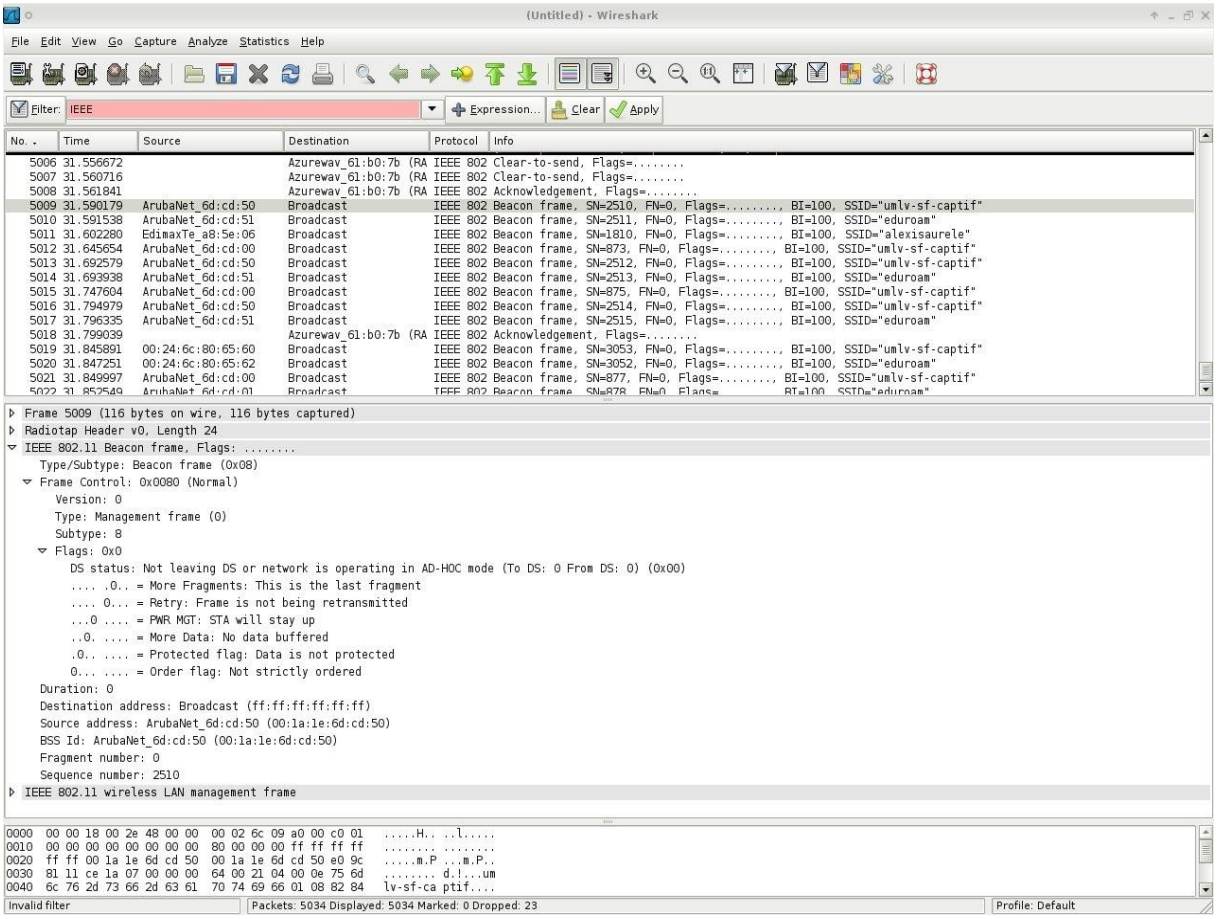
```
# iwconfig wlan0 essid “alexisaurele”
# iwconfig wlan0 channel 1
```

On vérifie ensuite que l'on utilise bien ce nom de réseau et la fréquence propre au *channel* que l'on a choisi.

```
# iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.

eth1      no wireless extensions.

wifi0     no wireless extensions.

wlan0      IEEE 802.11g  ESSID:"alexisaurele"
          Mode:Ad-Hoc  Frequency:2.442 GHz
          Bit Rate:0 kb/s
          RTS thr:off   Fragment thr:off
          Encryption key:off

          Link Quality=0/94  Signal level=-93 dBm  Noise
level=-93 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid
frag:0
          Tx excessive retries:0  Invalid misc:0   Missed
beacon:0

# iwlist wlan0 freq
wlan0      22 channels in total; available frequencies :
          Channel 01 : 2.412 GHz
          Channel 02 : 2.417 GHz
          Channel 03 : 2.422 GHz
          Channel 04 : 2.427 GHz
          Channel 05 : 2.432 GHz
          Channel 06 : 2.437 GHz
          Channel 07 : 2.442 GHz
          Channel 08 : 2.447 GHz
          Channel 09 : 2.452 GHz
          Channel 10 : 2.457 GHz
          Channel 11 : 2.462 GHz
          Current Frequency=2.442 GHz (Channel 1)
```

3. Les Beacon Frame (trames balises)

Nous analysons ensuite les trames (beacon frame) envoyées par cette station.



Les *beacon frames* servent à annoncer la présence d'une station émettrice dans le réseau. Un routeur écoute les trames balise pour reconnaître les nouveaux périphériques sans fils afin de s'associer et d'obtenir une adresse IP par exemple. On peut remarquer cela en regardant l'adresse destination qui est une adresse de diffusion (broadcast).

Voici la liste des informations contenues dans les *beacon frames* :

FC (*Frame Control*, en français *contrôle de trame*) : ce champ de deux octets est constitué des informations suivantes :

- **Version de protocole** : ce champ de 2 bits permettra de prendre en compte les évolutions de version du standard 802.11. La valeur est égale à zéro pour la première version.
- **Type** et **Sous-type** : ces champs, respectivement de 2 et 4 bits, définissent le type et le sous-type des trames. Le type *gestion* correspond aux demandes d'association ainsi qu'aux messages d'annonce du point d'accès. Le type *contrôle* est utilisé pour l'accès au média afin de demander des autorisations pour émettre. Enfin le type *données* concerne les envois de données (la plus grande partie du trafic).
- **To DS** : ce bit vaut 1 lorsque la trame est destinée au système de distribution (*DS*), il vaut zéro dans les autres cas. Toute trame envoyée par une station à destination d'un point d'accès possède ainsi un champ *To DS* positionné à 1.
- **From DS** : ce bit vaut 1 lorsque la trame provient du système de distribution (*DS*), il vaut zéro dans les autres cas. Ainsi, lorsque les deux champs *To* et *From* sont positionnés à zéro il s'agit d'une communication directe entre deux stations (mode *ad hoc*).
- **More Fragments** (*fragments supplémentaires*) : permet d'indiquer (lorsqu'il vaut 1) qu'il reste des fragments à transmettre
- **Retry** : à 1 ce bit spécifie que le fragment en cours est une retransmission d'un fragment précédemment envoyé (et sûrement perdu)
- **Power Management** (*gestion d'énergie*) : indique, lorsqu'il est à 1, que la station ayant envoyé ce fragment entre en mode de gestion d'énergie
- **More Data** (*gestion d'énergie*) : ce bit, utilisé pour le mode de gestion d'énergie, est utilisé par le point d'accès pour spécifier à une station que des trames supplémentaires sont stockées en attente.
- **WEP** : ce bit indique que l'algorithme de chiffrement WEP a été utilisé pour chiffrer le corps de la trame.
- **Order** (*ordre*) : indique que la trame a été envoyée en utilisant la classe de service strictement ordonnée (*Strictly-Ordered service class*)

Durée / ID : Ce champ indique la durée d'utilisation du canal de transmission.
Champs adresses : une trame peut contenir jusqu'à 3 adresses en plus de l'adresse de 48 bits
Contrôle de séquence : ce champ permet de distinguer les divers fragments d'une même trame. Il est composé de deux sous-champs permettant de réordonner les fragments :

- Le *numéro de fragment*
- Le *numéro de séquence*

4. Trames échangés au niveau LLC

Nous allons ensuite placer une deuxième station en mode ad-hoc avec la précédente.

Pour ce faire, nous configurons les adresses IP des deux stations :

station1 : 192.168.1.2

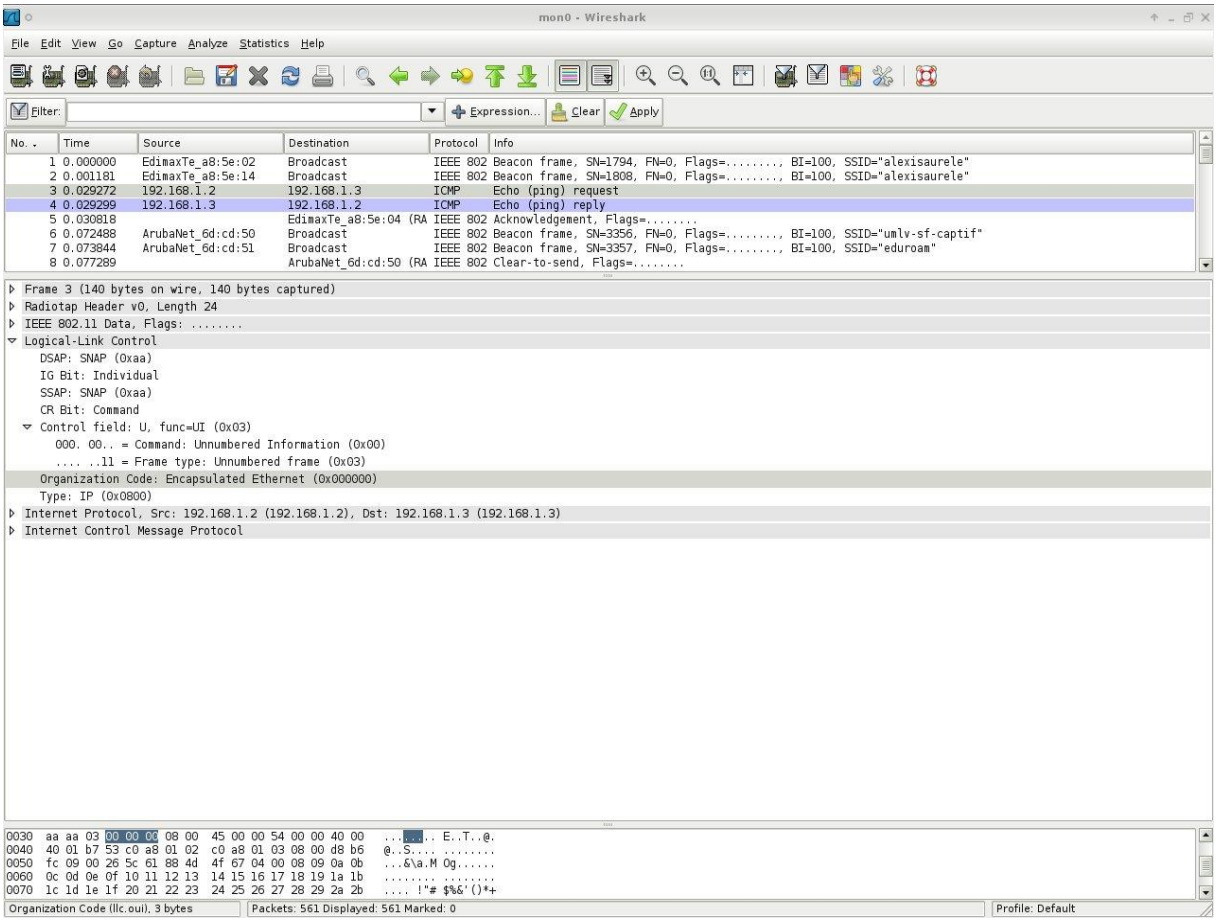
station2 : 192.168.1.3

```
pccop0b004-1:~# ping -c 1 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=3.27 ms

--- 192.168.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.270/3.270/3.270/0.000 ms
```

```
pccop0b004-3:~# ping -c 1 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=5.90 ms

--- 192.168.1.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 5.905/5.905/5.905/0.000 ms
```

Le ping est un succès, les deux stations communiquent. L'option **–c** indique qu'un seul paquet doit être envoyé.

Voici ce que nous obtenons sur la capture wireshark pendant les échanges.

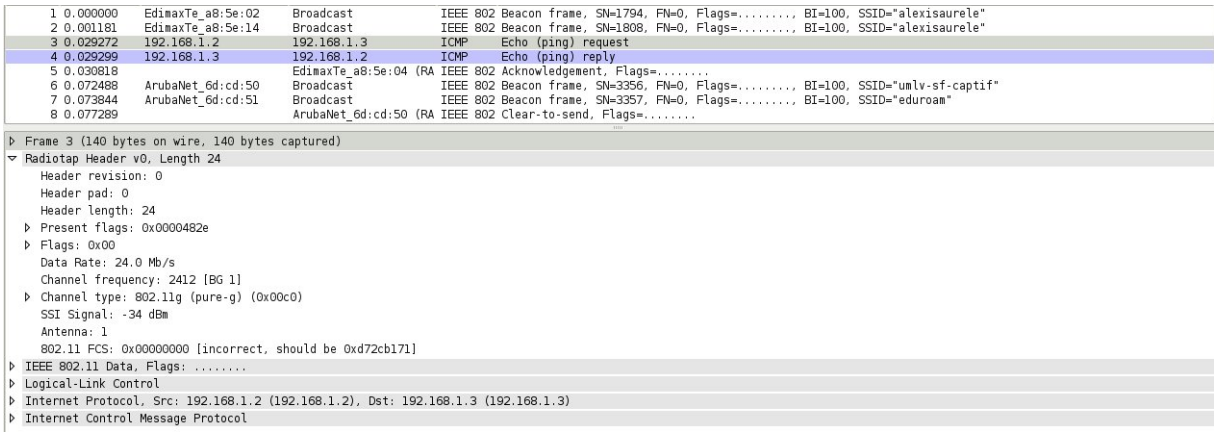


Les trames au niveau LLC (Logical Link Control) comportent les informations suivantes :

- DSAP (Destination Service Access Point) : adresse destination.
- IG Bit : adresse individuelle ou adresse de groupe.
- SSAP (Service Source Access Point) : adresse source.
- CR Bit : type de LLC, contrôle avec/sans connexion avec/sans acquittement.

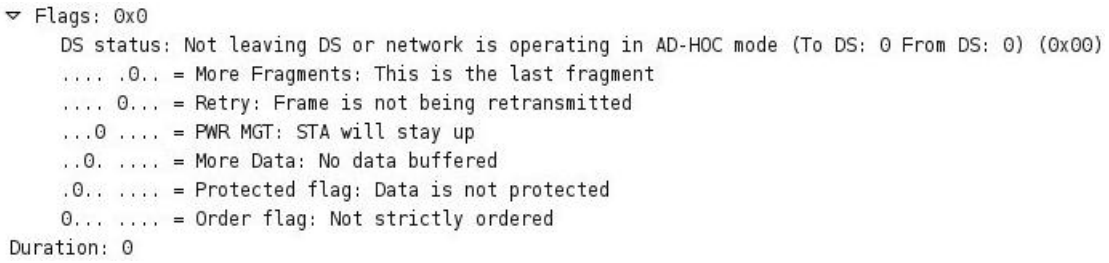
5. La couche Physique

Déterminez la valeur de débit à partir de l'entête de la couche physique.



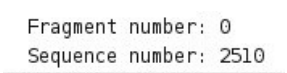
On peut voir que le débit est de 24 Mbits/s à partir de l'entête de la couche physique dans le champ *Date rate*.

Quelle est la valeur du champ Retry ?



La valeur du champ *Retry* est 0. Ce qui signifie que si un fragment se perd lors d'une transmission, ce dernier ne sera pas retransmit.

Est-ce que la trame est fragmentée au niveau de la couche 2 ? Quelle est la valeur du champ Duration ?



La trame peut être fragmentée au niveau de la couche 2. Dans ce cas la trame n'a pas été fragmentée car le flag MF (*More Fragment*) est à 0, le numéro de fragment (*Fragment number*) est à 0 et le champ *Duration*, qui indique la durée d'utilisation du canal de transmission est également à 0.

6. Technique CTS/RTS

Le principe est simple, la station voulant émettre écoute le réseau. Si le réseau est encombré, la transmission est différée. Dans le cas contraire, si le média est libre pendant un temps donné (appelé *DIFS* pour *Distributed Inter Frame Space*), alors la station peut émettre. La station transmet un message appelé *Ready To Send* (noté *RTS* signifiant *prêt à émettre*) contenant des informations sur le volume des données qu'elle souhaite émettre et sa vitesse de transmission. Le récepteur (généralement un point d'accès) répond un *Clear To Send* (*CTS*, signifiant *Le champ est libre pour émettre*), puis la station commence l'émission des données.

A réception de toutes les données émises par la station, le récepteur envoie un accusé de réception (*ACK*). Toutes les stations avoisinantes patientent alors pendant un temps qu'elle considère être celui nécessaire à la transmission du volume d'information à émettre à la vitesse annoncée.

La technique RTS/CTS est-elle activée par défaut sur les cartes ? Expliquez.

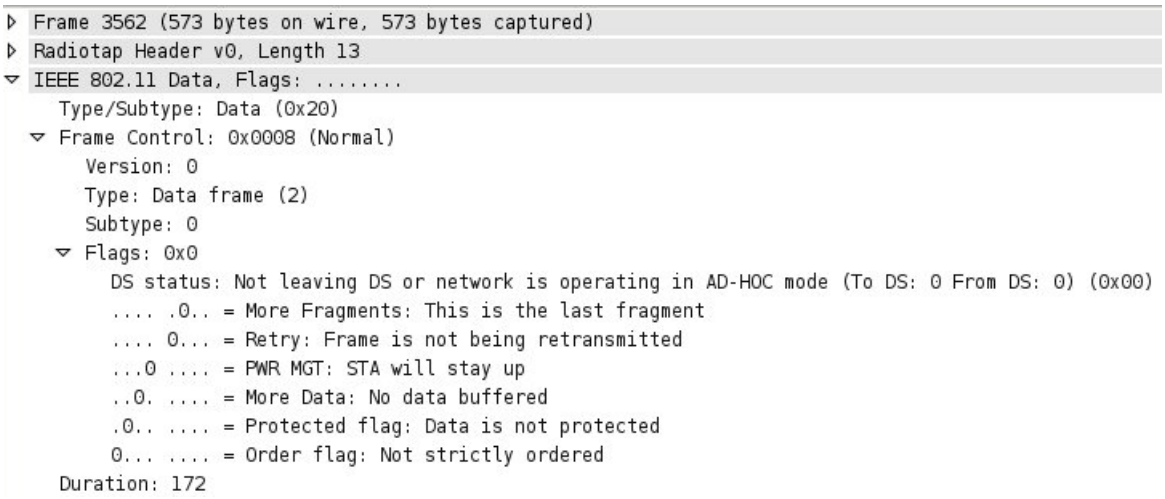
Le RTS/CTS n'est pas activé par défaut, car ces trames sont envoyées à bas débit (1Mbits/s) pour atteindre les périphériques les plus éloignées, or le temps de transmission de ces trames fait baisser le débit utile.

Nous positionnons le RTS à 500.

```
pccop0b004-1:~# iwconfig wlan0 rts 500
pccop0b004-1:~# iwconfig wlan0
wlan0     IEEE 802.11bg  ESSID:"alexisaurele"
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: D2:FF:9F:62:AB:01
          Tx-Power=4 dBm
          Retry  long limit:7   RTS thr=500 B   Fragment thr:off
          Encryption key:off
          Power Management:off
```

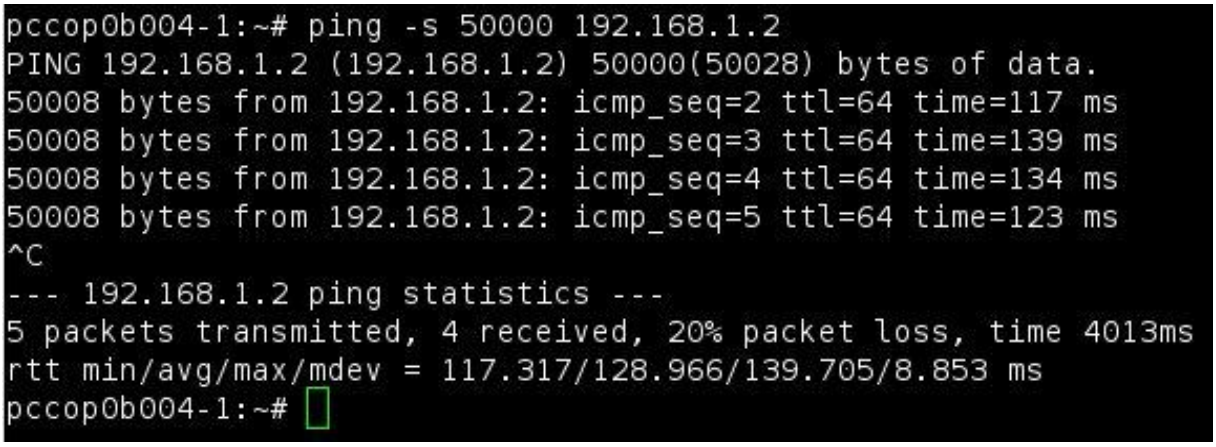
Nous effectuons un ping de 500 octets.

```
pccop0b004-1:~# ping -s 500 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 500(528) bytes of data.
508 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=3.90 ms
508 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=2.88 ms
508 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=2.98 ms
508 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=2.15 ms
^C
--- 192.168.1.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 2.156/2.982/3.909/0.626 ms
pccop0b004-1:~#
```



Le champ *Duration* est à 172. Ainsi la durée d'utilisation du canal de transmission est plus longue.

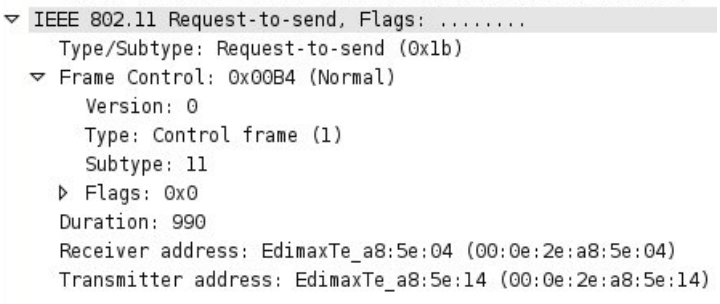
Maintenant nous réalisons un ping de 5Mo.



La trame est alors fragmentée, et *Duration* passe augmente de manière significative. Le canal de transmission accorde plus de temps pour échanger des données fragmentées.

Les trames RTS et CTS sont utilisées pour la réservation virtuelle des ressources dans le cadre de la procédure d'accès au support physique. Cette technique permet de diminuer les collisions lors des transmissions.
L'entête MAC comporte quelques différences suivant qu'il s'agisse de trames RTS, CTS ou ACK.

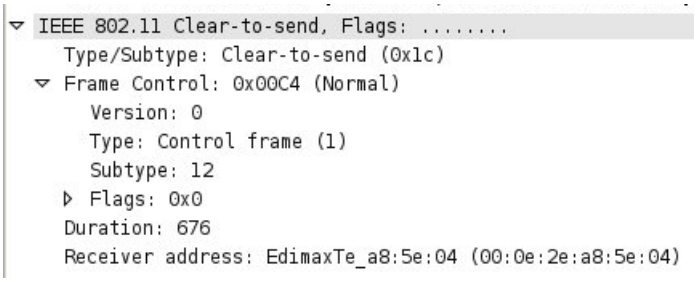
a) Trame RTS



L'entête de la trame RTS comprend les champs suivant :

- Frame Control : analogue au champ de la trame de données MAC.
- Duration : Durée à réserver.
- RA : Adresse de la station réceptrice.
- TA : Adresse de la station émettrice.

b) Trame CTS



L'entête de la trame CTS comprend les même champs que celui de RTS, hormis le champ TA. Le champ RA étant recopié à partir du champ TA de la trame RTS reçue.

On peut remarquer que l'adresse MAC source n'est pas présente, en effet la station qui envoie le CTS pour dire « j'écoute une station », peut être n'importe quelle station. On en vient donc au point faible de cette technique, imaginons qu'une machine envoie des CTS de manière rapide et régulière pour prendre toute la bande passante du réseau. Les périphériques sans fils deviennent alors vulnérables aux attaques et les hackers aiment cette technique de « CTS Flood » (inondation de CTS) car l'adresse MAC source n'est pas transmise. C'est une attaque par *déni de service* qui consiste à rendre indisponible un service, dans ce cas le réseau devient saturé.

En cas de fragmentation, un RTS/ CTS est envoyé avant chaque fragment ? Prouvez ça ou le contraire via une analyse par wireshark.

3063	12.564386	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3064	12.565295	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=29600)
3065	12.565668	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3066	12.566843	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=31080)
3067	12.567217	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3068	12.570069		00:25:d3:c2:ef:81 (RA)	IEEE 802	Clear-to-send, Flags=........
3069	12.571846	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=32560)
3070	12.572220	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3071	12.573108	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=34040)
3072	12.573480	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3073	12.574450	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=35520)
3074	12.574823	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3075	12.575673	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=37000)
3076	12.576045	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3077	12.576814	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=38480)
3078	12.577188	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3079	12.578115	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=39960)
3080	12.578489	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3081	12.579277	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=41440)
3082	12.579650	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3083	12.580460	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=42920)
3084	12.580833	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3085	12.581803	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=44400)
3086	12.582174	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3087	12.583254	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=45880)
3088	12.583628	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3089	12.584497	192.168.1.2	192.168.1.3	IP	Fragmented IP protocol (proto=ICMP 0x01, off=47360)
3090	12.584871	EdimaxTe_a8:5e:14 (TA	EdimaxTe_a8:5e:04 (RA)	IEEE 802	Request-to-send, Flags=........
3091	12.585537	192.168.1.2	192.168.1.3	ICMP	Echo (ping) reply
3092	12.590944		00:25:d3:c2:ef:81 (RA)	IEEE 802	Clear-to-send, Flags=........

D'après la capture wireshark ci-dessus, il s'avère qu'un RTS est envoyé avant chaque fragment justement pour éviter les collisions de transmissions. Cependant les trames CTS sont envoyées moins souvent mais de manière périodique, probablement proportionnelle au champ *Duration*.

En changeant le débit de la carte, montrez que ce débit est transmis au niveau de l'entête de la couche physique.

Nous changeons le débit de l'interface wlan0 à 11 Mb/s.

```
pccop0b004-1 :~# iwconfig wlan0 rate 11M
```

```
▷ Frame 3 (140 bytes on wire, 140 bytes captured)
▽ Radiotap Header v0, Length 24
    Header revision: 0
    Header pad: 0
    Header length: 24
  ▷ Present flags: 0x0000482e
  ▷ Flags: 0x00
    Data Rate: 11.0 Mb/s
    Channel frequency: 2412 [BG 1]
  ▷ Channel type: 802.11g (pure-g) (0x00c0)
    SSI Signal: -34 dBm
    Antenna: 1
    802.11 FCS: 0x00000000 [incorrect, should be 0xd72cb171]
```

Ainsi, le champ *Data Rate* est bien transmis dans l'entête de la couche physique.

Quelle est la valeur par défaut de nombre de retransmissions de votre carte ?

```
pccop0b004-1:~# iwconfig wlan0
wlan0     IEEE 802.11bg  ESSID:"alexisaurele"
          Mode:Ad-Hoc  Frequency:2.412 GHz  Cell: D2:FF:9F:62:AB:01
          Tx-Power=4 dBm
          Retry  long limit:7   RTS thr=500 B   Fragment thr:off
          Encryption key:off
          Power Management:off
```

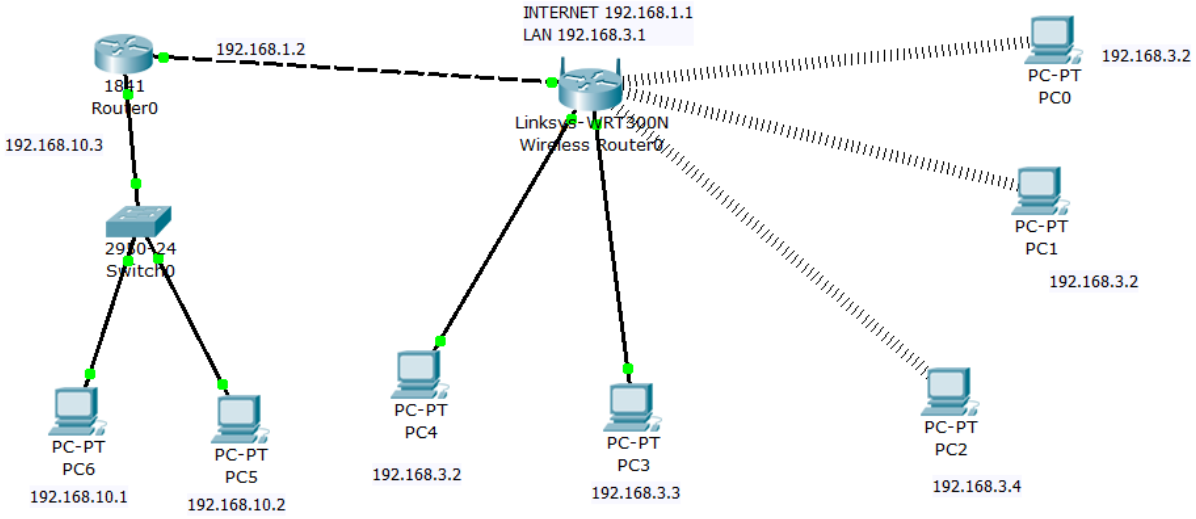
Avec la commande *iwconfig (lorsque RTS est activé)*, le champ *Retry long limit* indique le nombre de retransmissions de l'interface *wlan0*.

Comment peut-on modifier cette valeur ?

Cette valeur est modifiable avec ma commande `iwconfig wlan0 retry 10` pour passer à 10 retransmissions maximum.

7. WiFi sous Packet tracer

Packet tracer est un logiciel de simulation de réseaux virtuels créée par cisco. Nous avons configuré PC0, PC1, PC2, PC3 et PC4 en IP dynamique, PC5 et PC6 possèdent des adresses statiques.



La connexion fonctionne.

Attention !

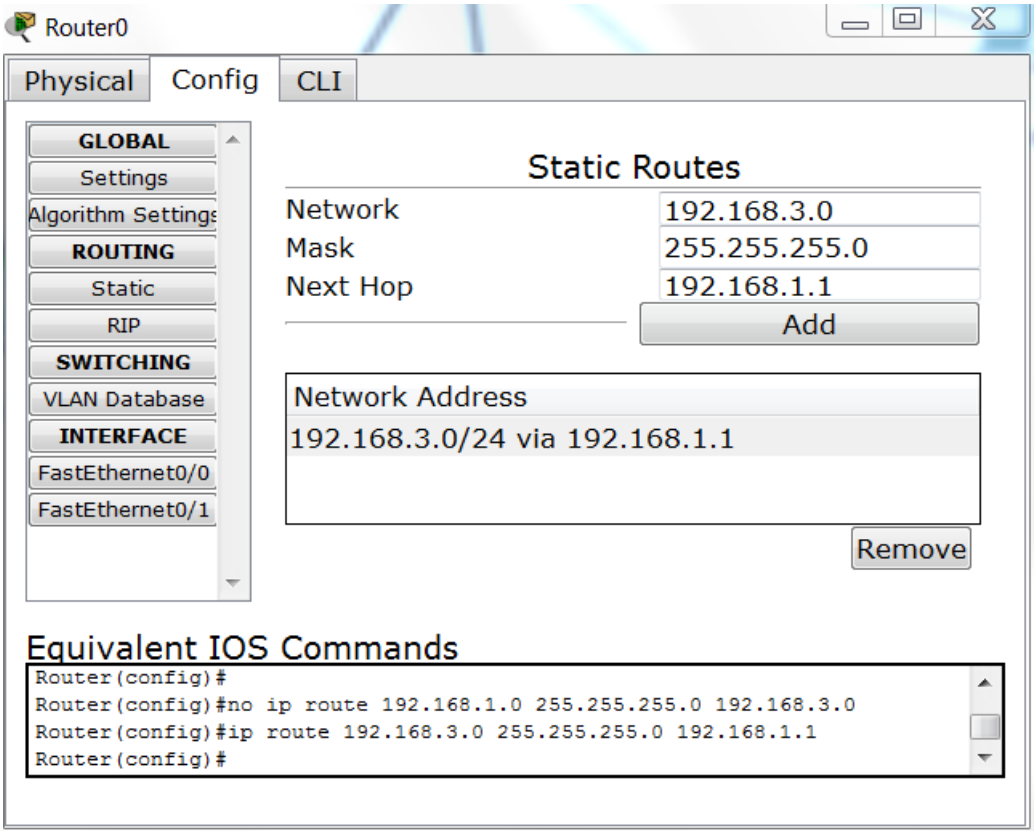
Veuillez à mettre sur ON les ports ethernet du switch.

Nous effectuons un ping à partir de PC5 vers PC2.

```
PC>ping 192.168.3.4

Pinging 192.168.3.4 with 32 bytes of data:

Reply from 192.168.10.3: Destination host unreachable.
Reply from 192.168.10.3: Destination host unreachable.
Reply from 192.168.10.3: Destination host unreachable.
Reply from 192.168.10.3: Destination host unreachable.

Ping statistics for 192.168.3.4:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Sans succès, la machine est inaccessible car elle n'appartient pas au même réseau que la machine source (source : 192.168.10.3, destination : 192.168.3.4).

Pour remédier à cela nous ajoutons une route sur le routeur.



De cette manière les paquets en direction du routeur sont redirigés sur le réseau local appartenant au routeur sans fil.

```
PC>ping 192.168.3.4

Pinging 192.168.3.4 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time=26ms TTL=126
Reply from 192.168.1.1: bytes=32 time=22ms TTL=126
Reply from 192.168.1.1: bytes=32 time=26ms TTL=126
Reply from 192.168.1.1: bytes=32 time=17ms TTL=126

Ping statistics for 192.168.3.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 17ms, Maximum = 26ms, Average = 22ms
```

Et les deux stations peuvent maintenant communiquer correctement.

Conclusion

Nous arrivons à la fin de ce TP avec de meilleures connaissances dans le monde du WiFi. Nous avons étudié en détails le principe CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) composé de trames RTS (*Ready-To-Send)* et CTS (*Clear-To-Send)* et d'acquittement (ACK) permettant de diminuer le risque de collisions des paquets transmis. D'autre part nous avons pu découvrir Packet Tracer qui se montre être une application très utile et efficace pour simuler des configurations de réseaux divers et variés.