Le protocole ICMP appartient a la couche 3 du modèle OSI, c'est à dire la couche Réseau.

Nous avons bien des trames ARP car nous avons vidé la table ARP avant d'effectuer le ping. (la commande a faire : arp -d*) (requête « who has 172,16,255,254 ? » et la réponse arp avec l'adresse mac qui correspond)

```
C:\Users\Admin>ping r2-central.example.com

Envoi d'une requête 'ping' sur r2-central.example.com [172.16.255.254] avec 32 octets de données :
Réponse de 172.16.255.254 : octets=32 temps=1 ms TTL=255
Réponse de 172.16.255.254 : octets=32 temps=1 ms TTL=255
Réponse de 172.16.255.254 : octets=32 temps=1 ms TTL=255
Réponse de 172.16.255.254 : octets=32 temps<1ms TTL=255

Statistiques Ping pour 172.16.255.254:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 1ms, Moyenne = 0ms

C:\Users\Admin>
```

```
C:\Users\Admin>arp -d*

Affiche et modifie les tables de traduction d'adresses IP en adresses
physiques utilisées par le protocole de résolution d'adresses ARP.

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

  -a            Affiche les entrées ARP en cours en interrogeant les données
                en cours du protocole. Si inet_addr est spécifié, seules les
                adresses IP et physiques de l'ordinateur spécifié sont
                affichées. Si plus d'une interface réseau utilise ARP, les
                entrées de chaque table ARP sont affichées.
  -g            Identique à -a.
  -v            Affiche les entrées ARP en cours en mode verbeux. Toutes les
                entrées non valides ainsi que celles de l'interface de retour
                de bouclage sont affichées.
  inet_addr     Spécifie un adresse Internet.
  -N if_addr    Affiche les entrées ARP de chaque interface réseau spécifiée
                par if_addr.
  -d            Supprime l'hôte spécifié par inet_addr. inet_addr peut
                contenir le caractère générique * pour supprimer tous
                les hôtes.
  -s            Ajoute l'hôte et associe l'adresse Internet inet_addr
                avec l'adresse physique eth_addr. L'adresse physique
                est donnée sous forme de 6 octets hexadécimaux séparés
                par des tirets. L'entrée est permanente.
  eth_addr      Spécifie une adresse physique.
  if_addr       Spécifie l'adresse Internet de l'interface dont la table
                de traduction d'adresses doit être modifiée.
                Si ce paramètre n'est pas indiqué, la première interface
                applicable sera utilisée.
Exemples :
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Ajoute une entrée statique.
  > arp -a                                    .... Affiche la table ARP.
```

Le nom « r2-central.example.com » apparaît dans les trames car avant d'effectuer le ping, le système a besoin de résoudre ce nom de domaine en adresse IP. On observe dans la capture : une requête DNS de notre machine vers le serveur DNS pour demander l'adresse IP associée à "r2-central.example.com", puis le DNS fournit l'adresse IP correspondante et ensuite, les paquets ICMP sont envoyés vers cette adresse IP

icmp

No.	Time	Source	Destination	Protocol	Lengtl	Info
250	16.641357	172.16.254.14	192.168.254.254	ICMP	74	Echo (ping) request id=0x0001, seq=5/1280, ttl=128 (reply in 251)
251	16.643238	192.168.254.254	172.16.254.14	ICMP	74	Echo (ping) reply id=0x0001, seq=5/1280, ttl=62 (request in 250)
255	17.652274	172.16.254.14	192.168.254.254	ICMP	74	Echo (ping) request id=0x0001, seq=6/1536, ttl=128 (reply in 256)
256	17.654970	192.168.254.254	172.16.254.14	ICMP	74	Echo (ping) reply id=0x0001, seq=6/1536, ttl=62 (request in 255)
257	18.668347	172.16.254.14	192.168.254.254	ICMP	74	Echo (ping) request id=0x0001, seq=7/1792, ttl=128 (reply in 258)
258	18.670599	192.168.254.254	172.16.254.14	ICMP	74	Echo (ping) reply id=0x0001, seq=7/1792, ttl=62 (request in 257)
260	19.687018	172.16.254.14	192.168.254.254	ICMP	74	Echo (ping) request id=0x0001, seq=8/2048, ttl=128 (reply in 261)
261	19.688887	192.168.254.254	172.16.254.14	ICMP	74	Echo (ping) reply id=0x0001, seq=8/2048, ttl=62 (request in 260)

Le service DNS qui est utilisée

Le client envoi une requête DNS de type (généralement en UDP) qui contient un nom de domaine et le serveur DNS répond avec l'adresse IP associée

DNS utilise le protocole UDP port 53 (standard pour DNS). Le DNS utilise l'unicast (communication point à point)

```
C:\Windows\System32>scapy
←[39mINFO: Can't import PyX. Won't be able to use psdump() or pdfdump().←[0m
←[33m←[1mWARNING: Wireshark is installed, but cannot read manuf !←[0m←[0m
←[39mINFO: Can't import python-cryptography v1.7+. Disabled PKI & TLS crypto-related features.←[0m
←[39mINFO: Can't import python-cryptography v1.7+. Disabled WEP decryption/encryption. (Dot11)←[0m
←[39mINFO: Can't import python-cryptography v1.7+. Disabled IPsec encryption/authentication.←[0m
←[33m←[1mWARNING: IPython not available. Using standard Python shell instead.
AutoCompletion, History are disabled.←[0m←[0m
←[33m←[1mWARNING: On Windows, colors are also disabled←[0m←[0m

                     aSPY//YASa
             apyyyyCY//////////YCa       |
            sY//////YSpcs  scpCY//Pp     | Welcome to Scapy
 ayp ayyyyyyySCP//Pp           syY//C    | Version 2.5.0
 AYAsAYYYYYYYY///Ps              cY//S   |
         pCCCCY//p          cSSps y//Y   | https://github.com/secdev/scapy
         SPPPP///a          pP///AC//Y   |
              A//A            cyP////C   | Have fun!
              p///Ac            sC///a   |
              P////YCpc           A//A   | Craft packets like I craft my beer.
       scccccp///pSP///p          p//Y   |               -- Jean De Clerck
      sY/////////y  caa           S//P   |
       cayCyayP//Ya              pY/Ya
        sY/PsY////YCc          aC//Yp
         sc  sccaCY//PCypaapyCP//YSs
                  spCPY//////YPSps
                       ccaacs

>>> eth=Ether()
>>> ip=IP(dst="192.168.254.254")
>>> icmp=ICMP()
>>> packet=eth/ip/icmp
>>> ans, unans=srp(packet)
Begin emission:
Finished sending 1 packets.
...*
Received 4 packets, got 1 answers, remaining 0 packets
>>> ans.summary()
Ether / IP / ICMP 172.16.254.14 > 192.168.254.254 echo-request 0 ==> Ether / IP / ICMP 192.168.254.254 > 172.16.254.14 echo-reply 0 / Padding
>>>
```

Les commandes utilisées sont :

eth = Ether()

ip = IP(dst="192.168.254.254")

icmp = ICMP()

packet = eth/ip/icmp

ans, unans = srp(packet)

ans.summary()