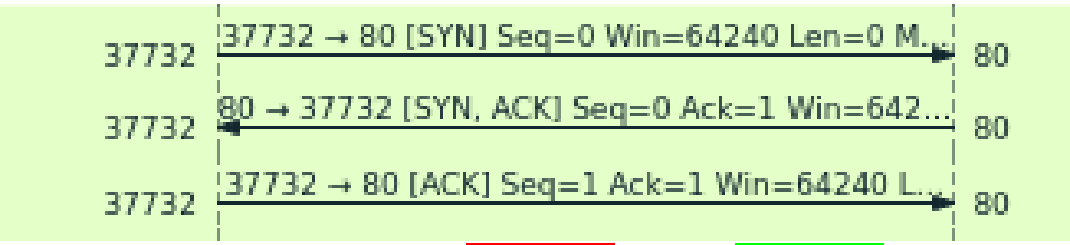
1. TCP, http, DNS
2. dns.qry.name == "manpages.org" ou dns contains "manpages"
3. MAC SRC = 00:0c:29:a9:0c:04 IP SRC = 192.168.254.5
4. Protocole 0x0800 ➔ IPV4
5. MAC DST = 00:50:56:e8:ab:c3 IP DST = 9.9.9.9
6. La MAC DST est celle de ma passerelle par défaut (default gateway, default route, etc.)
   L'adresse IP DST est celle du DNS SERVEUR DE GOOGLE (9.9.9.9)
7. La taille du header IP est de 20o et la taille totale du paquet est de 58o
8. Le protocole de la couche transport est de l'UDP (17 ou 0x11)
9. Le port UDP SRC = 46592 et le port UDP DST = 53 . Le port SRC est supérieur à 49152-65535 et est dynamique (il change pour tous). Le port DST 53 signifie que la couche application est du DNS.
10. La longueur de l'entête UDP est de 38o, elle correspond bien à la taille du paquet IP de 58o total (soit 38o UDP + 20o de l'entête IP).
11. C'est le flag qui renseigne le type de message, le dernier bit du flag à 0 = Requête et 1 = Réponse.
12. Type A et classe IN
13. Transaction ID: 0x6eda
14. MAC DST = 00:0c:29:a9:0c:04 IP DST = 192.168.254.5
    et MAC SRC = 00:50:56:e8:ab:c3 IP SRC = 9.9.9.9
15. L'entête IP est toujours de 20o, mais la taille du paquet est de 106o pour la réponse, contre 58o pour la requête. Parce que la réponse DNS est contenue dans les DATA.
16. Transaction ID: 0x6eda, sont identiques
17. 3 IP en réponse, TTL est géré par l'administrateur du DNS qui renseigne la durée de vie maximum de l'information dans le système de cash du DNS client.
18.

Application	DNS	donnée/data	
Transport	UDP	segment	port 53
Internet	IP	packet	
Accès réseau	Ethernet	Frame/trame	

19. MAC SRC = 00:0c:29:a9:0c:04 IP SRC = 192.168.254.5
    MAC DST = 00:50:56:e8:ab:c3 != IP DST = 104.27.164.196
    TCP
    IP
    Ethernet
20. port DST : 80 → protocole http port SRC > 1024 et est dynamique
21. SEQ (0 en relatif) 1488354669 et le MSS = 1460o
22. SYN flag → Synchronisation des numéros de séquences
23. SEQ (0 en relatif) **1188843366** et le MSS = 1460o
24. ACK → Accusé de réception avec le numéro de SEQ du client + 1 → 1488354669 + 1 **1488354670**
25. Pour valider la bonne réception du segment 1488354669, donc le serveur répond tu peux passer au segment suivant, c'est ce qu'on appelle le segment initial 1488354670.
26. ACK → Le client envoie l'accusé de réception du numéro de séquence du serveur
    SEQ 1488354670 ACK **1188843366 + 1** → **1188843367**

27. Les numéros de séquences initiaux client et serveur sont établis et connus sur le client et le serveur. La communication http peut se réaliser à partir de ces numéros de séquence TCP.
28. 

29. Le GET http client vers le serveur : SEQ 1488354670 et le ACK **1188843367**
30. Le flag PUSH et ACK pour envoyer la requête http
31. L'entête TCP fait 20o et 418o de charge (de données http du niveau application)
32.
33. Le SEQ = 1 (**1188843367**) et ACK = 418 +1488354670 = 1488355088
34. Effectivement le serveur répond avec un TCP ACK et pas un http (la couche application se repose sur la couche transport qui assure la session TCP et la bonne réception des données)
35. Le SEQ = 1 (**1188843367**)
36. **19301o**, ou 19,301Ko de charge
37. PUSH + ACK
38. Le SEQ = 1488355088 et l'ACK **1188843367 + 19301 + 17** = **1188862685**
39. http code 200 OK
40. Le code HTML est identique au niveau du trafic réseau et du navigateur WEB
41.
    a. Requête DNS → NOM = IP
    b. Etablissement de la session TCP (3 ways handshake)
    c. L'échange http qui se fait entre le client et le serveur WEB