TP - Redécouverte de plusieurs protocoles

17 novembre 2022

On réalisera un rapport écrit avec Word, ou un éditeur LaTeX, exporté en PDF.

1. Ouvrir un navigateur (Firefox ou Google Chrome de préférence), puis lancer leur inspecteur.
2. Ouvrir Wireshark et lancer une écoute.
3. Ouvrir https://random.dog/woof.json, puis avec l'adresse obtenue, lancer la requête pour obtenir une photo d'un chien aléatoire. Lorsque la photo est chargée, mettre fin à l'écoute et enregistrer le fichier Wireshark.
4. Quels sont les protocoles mis en jeu pour obtenir la page ?
5. Indiquer le protocole dont le paquet contient l'adresse mail du serveur. Retracer l'échange de votre requête jusqu'à la réponse.
6. Faire une capture d'écran des octets (donnés en hexadécimal) du premier segment TCP émis. Mettre en évidence sur l'image les données des autres protocoles encapsulés à l'intérieur. En s'aidant de la fenêtre d'étude des protocoles de Wireshark, donner leur nom, vérifier et indiquer votre adresse IP ainsi que le port utilisé pour la connexion.
7. Indiquer le nombre total de segments de la transaction.

1 Protocole IP

Nous détaillons les informations du schéma de la figure 1.

— Le champ Version (4 bits) indique le numéro de version du protocole IP utilisé. Les valeurs les plus courantes sont 4 et 6 pour IPv4 et IPv6.

— Le champ IHL (4 bits) contient la taille (i.e. le nombre de mots de 32 bits) de cet en-tête. 5 minimum, mais peut prendre des valeurs jusqu'à 15.

— Le champ Service (8 bits) permet de gérer la qualité de service de la couche 3 du modèle OSI.

— Le champ Longueur Totale (16 bits) contient un entier indiquant la longueur du paquet en octets, en incluant l'en-tête et les données contenues.

— Le champ Identification (16 bits) sert à identifier le paquet fragmenté pour que le destinataire puisse ordonner les paquets reçus. On rappelle que l'Ethernet limite la taille maximale des paquets à 1500 octets.

— Les champs O, DF, MF (1 bit chacun) renseignent l'état de fragmentation :

  — Le champ O vaut toujours 0 ;

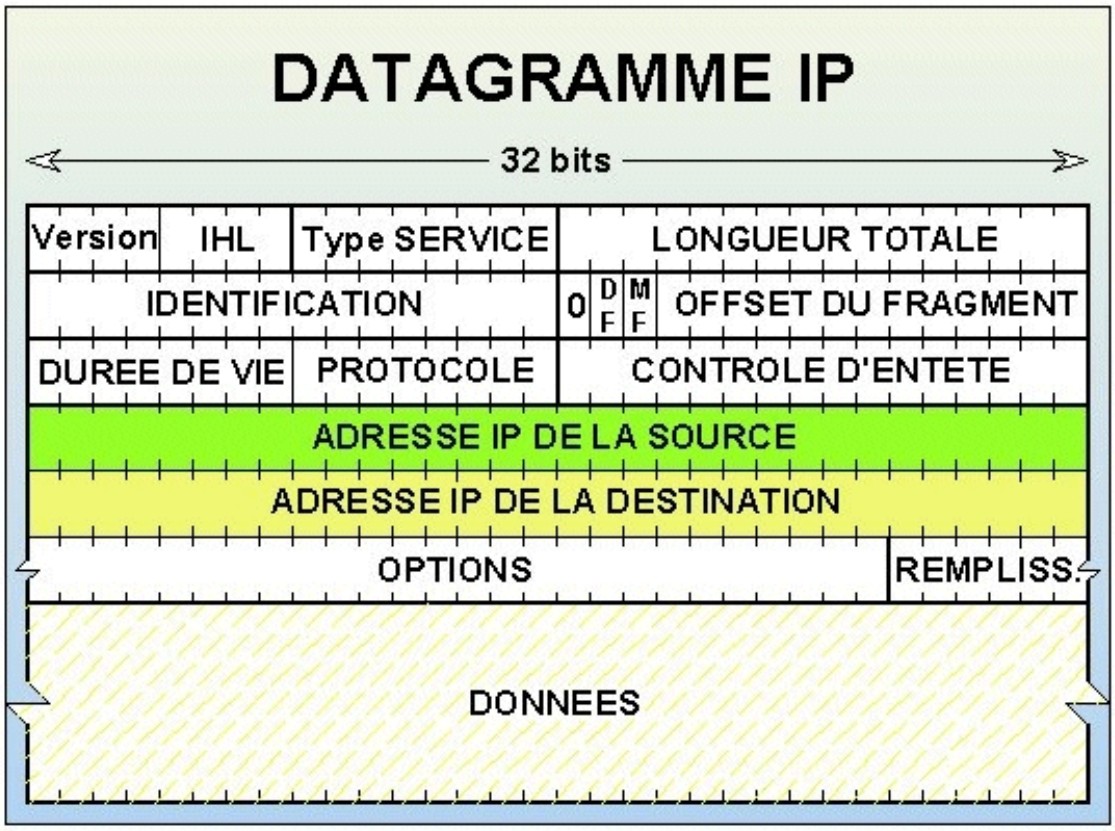

FIGURE 1 – Contenu de l'en-tête IP
Source: http ://arsene.perez-mas.pagesperso-orange.fr/reseaux/tcpip/entete_ip.htm

  - DF (***D**on't **F**ragment*) indique si le paquet peut-être fragmenté ;
  - MF (***M**ore **F**ragments*) indique si le paquet est le dernier.
- Le champ Offset (13 bits) contient un entier représentant le décalage entre le premier octet de données du datagramme non-fragmenté et le premier octet des données fragmentées qu'il transporte. Plus précisément, cette valeur représente le nombre de mots de 8 octets et non le nombre d'octets.

  Ainsi, dans l'exemple proposé par la figure 2, on souhaite envoyer un paquet de 10000 octets au destinataire. Cependant, la connexion entre les routeurs R1 et R2 a un MTU de 4000. Le paquet initial est alors scindé en 3 fragments :
  - Le premier a un offset de 0 (c'est le premier), et le champ MF à 1 car il y a encore des fragments ;
  - Le deuxième a un offset de 497, car 3976 octets ont été transmis, soit 497 mots de 8 octets ;
  - Le dernier fragment a donc son champ MF à 0.

  Comme la liaison entre le routeur R2 et le destinataire a un MTU inférieur à 4000, ces fragments sont à nouveau divisés. La division s'effectue de telle sorte à pouvoir reconstituer les fragments précédents.
- Le champ Durée de Vie (***T**ime **t**o **l**ive* en anglais) (8 bits) indique le nombre de transferts restants du paquet avant d'être détruit. Si le paquet reste dans la file

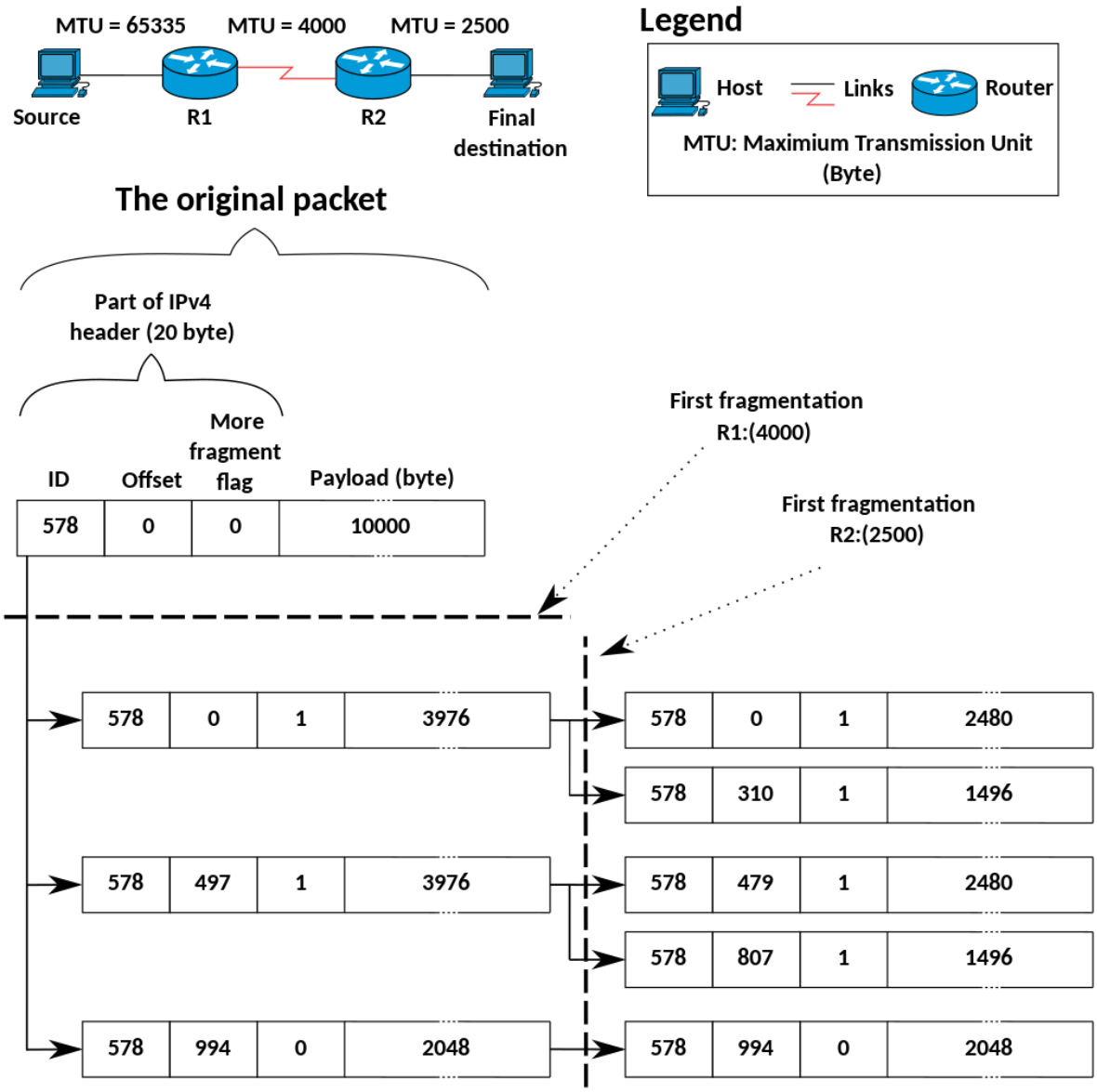

FIGURE 2 – Cas où la fragmentation s'avère nécessaire : la taille du paquet est supérieure à la limite fixée par la liaison.
Source: https ://en.wikipedia.org/wiki/IP_fragmentation (Auteur : Michel Bakni)

d'attente du routeur plus d'une seconde, alors chaque seconde passée fait décrémenter ce compteur. Cela évite par exemple de transférer des paquets pris dans des boucles.

— Le champ Protocole (8 bits) indique le code du protocole porté par le datagramme. On aura les valeurs 6 pour le protocole TCP et 17 pour le protocole UDP.

— Le champ Contrôle d'en-tête (16 bits) qui vérifie l'intégrité des données transmises. Il peut être un CRC.

— Les champs IP source et IP Destination (32 bits chacun) contiennent les adresses IP nécessaires au transfert.

2 Protocole TCP

TCP est un protocole de communication fiable à flot d'octets orienté connexion.

La connexion entre un client et un serveur s'effectue avec une ouverture en trois temps (*three-way handshake*). Le point de connexion d'un serveur, nommé *socket*, attend alors une demande de connexion d'un client.

1. Le client, souhaitant se connecter, envoie un segment SYN au serveur qui contient le numéro de séquence à utiliser comme numéro d'initialisation.
2. Le serveur retourne message contenant le message SYN du client, et un accusé de réception ACK.
3. Le client commence à envoyer des données, jusqu'à terminaison de la liaison, qui s'effectue avec un *three-way handshake* contenant le drapeau FIN qui signifie que l'émetteur n'a plus de données à transmettre.

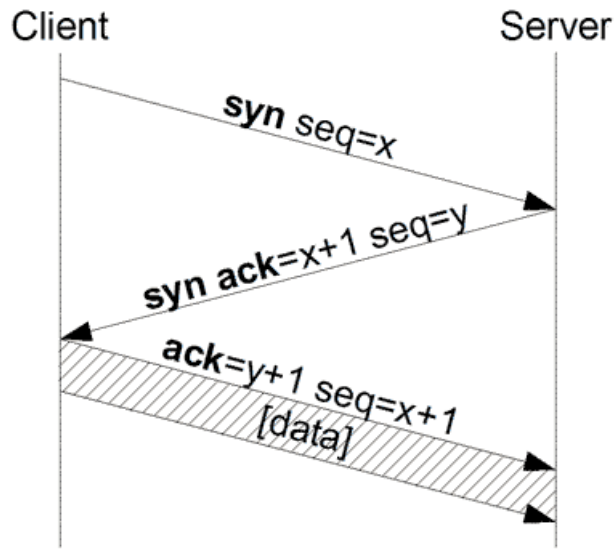

FIGURE 3 – Schéma temporel du Three-way handshake
Source: https ://commons.wikimedia.org/wiki/File :Tcp-handshake.png

2.1 Mécanisme d'acquittement

Le premier paquet envoyé par le client est marqué par un nombre de séquence (SEQ) aléatoire x, qu'on ramènera à 1 pour faciliter la compréhension. Puis ce nombre augmente du nombre d'octets qui ont été émis : si le client émet n octets, son prochain paquet aura pour numéro SEQ $x+n$. Du côté du serveur, celui-ci accuse réception d'un certain nombre d'octets. En recevant les n octets et la valeur initiale x, le serveur retourne la valeur $x+n$ dans le paramètre ACK, indiquant que les n octets ont été reçus.

Ce mécanisme est cumulatif : la valeur de ACK détermine le numéro SEQ attendu pour le prochain paquet. Si on reçoit un segment avec un numéro SEQ plus grand que celui attendu, on le conserve, mais sans l'acquitter, on attend d'avoir reçu tous les segments pour le faire. Si un segment n'est pas acquitté, il est considéré comme perdu et il doit être retransmis (la transmission reprend au dernier octet acquitté, donc les éventuels segments qui ont quand même été reçus seront détruits).

2.2 Contrôle du flux

Chaque machine communique la taille de sa mémoire tampon à son interlocuteur, qu'on appelle fenêtre. Elle correspond à la quantité d'information qu'on peut envoyer sans attendre acquittement. Si l'interlocuteur a une fenêtre de 32000 octets, alors il est possible d'émettre jusqu'à 32000 octets sans se préoccuper de la réception.

FIGURE 4 – Illustration du principe de la fenêtre TCP

2.3 Retransmission

On rappelle que le *round time trip*, qu'on notera RTT par la suite, est le temps entre l'émission d'un segment et son acquittement.

Le temps de temporisation avant retransmission doit être choisi avec soin :

— s'il est trop court, on risque d'envoyer beaucoup de paquets inutiles ;

— s'il est trop long, alors la durée de transmission risque d'augmenter fortement.

L'algorithme de Jacobson permet de modifier dynamiquement ce temps. On évalue le RTT à l'aide de timestamps, et on le pondère par un coefficient a compris entre 0 et 1, ainsi que l'ancien RTT estimé pour le calcul :

$$RTT_{n+1} = a \times RTT_n + (1 - a)RTT_{nv} \tag{1}$$

En général, on prend a proche de 0,9.

On calcule ensuite la temporisation à partir d'un coefficient $b > 1$, mais qui doit rester assez proche de la valeur réelle.

$$T = b \times RTT_{n+1} \tag{2}$$

L'idée est de rendre b variable : à chaque fois qu'un nouveau ACK est reçu, on calcule l'écart du nouveau RTT avec le RTT estimé (Δt), pondéré par a :

$$D = (1 - a) \times \Delta t \tag{3}$$

La plupart des machines exploitent cette valeur de T :

$$T = RTT \times 4D \tag{4}$$

2.4 Détail de la trame

On détaillera ci-dessous les éléments d'une trame TCP (voir figure 5). Son en-tête contient au minimum 20 octets, mais peut augmenter avec des éléments optionnels.

- Les champs Port source et Port destination (16 bits chacun) représentent respectivement le numéro d'application de l'expéditeur et le numéro d'application du destinataire.
- Le champ Numéro de séquence (32 bits) contient le nombre d'octets envoyés par la machine émettrice.
- Le champ Numéro d'acquittement (32 bits) contient le nombre d'octets qui ont été reçus de la part de l'autre machine.
- Le champ Offset (4 bits) indique le numéro du mot de la trame à partir duquel les données de l'application débutent. Ce champ contient très souvent la valeur 5 car un en-tête sans option fait 20 octets.
- Le champ Réservé (6 bits) est, comme son nom l'indique, présent dans le cas où on souhaite rajouter un ou plusieurs champs. Ses 3 premiers bits sont devenus (RFC 3168 / RFC 3540) un champ ECN ou NS, signalant la présence de congestion
- Le champ de Contrôle (6 bits), contenant les données suivantes :
  - URG : un pointeur de données urgentes ;
  - ACK : indique que le paquet est un accusé de réception ;
  - PSH : indique que les données sont à envoyer tout de suite ;
  - RST : indique une rupture anormale de la connexion ;
  - SYN : indique une demande de synchronisation ;
  - FIN : indique une demande de fin de connexion.
- Le champ Fenêtre (16 bits) indique le nombre d'octets disponibles dans la mémoire tampon de réception ;
- Le champ Total de contrôle (*Checksum*) (16 bits) calculé sur l'ensemble de l'en-tête ;
- Le champ Position d'urgence (16 bits) qui indique le nombre d'octets, en partant du premier, qui sont considérées comme des données à transmettre de manière urgente (en premier)

Nous laisserons de côté le champ Options et le remplissage.

La figure 7 présente une trame TCP et plus particulièrement la partie *flags*, où celui de la synchronisation est à 1. On prendra soin de ne pas envoyer de données compromettantes claires, comme décrit dans la figure 8.

La fiabilité vient de l'utilisation du PAR (*Positive Acknowlegment with Retransmission*) qui consiste à retransmettre un **segment** dont on n'a pas reçu l'accusé de réception après un certain temps.

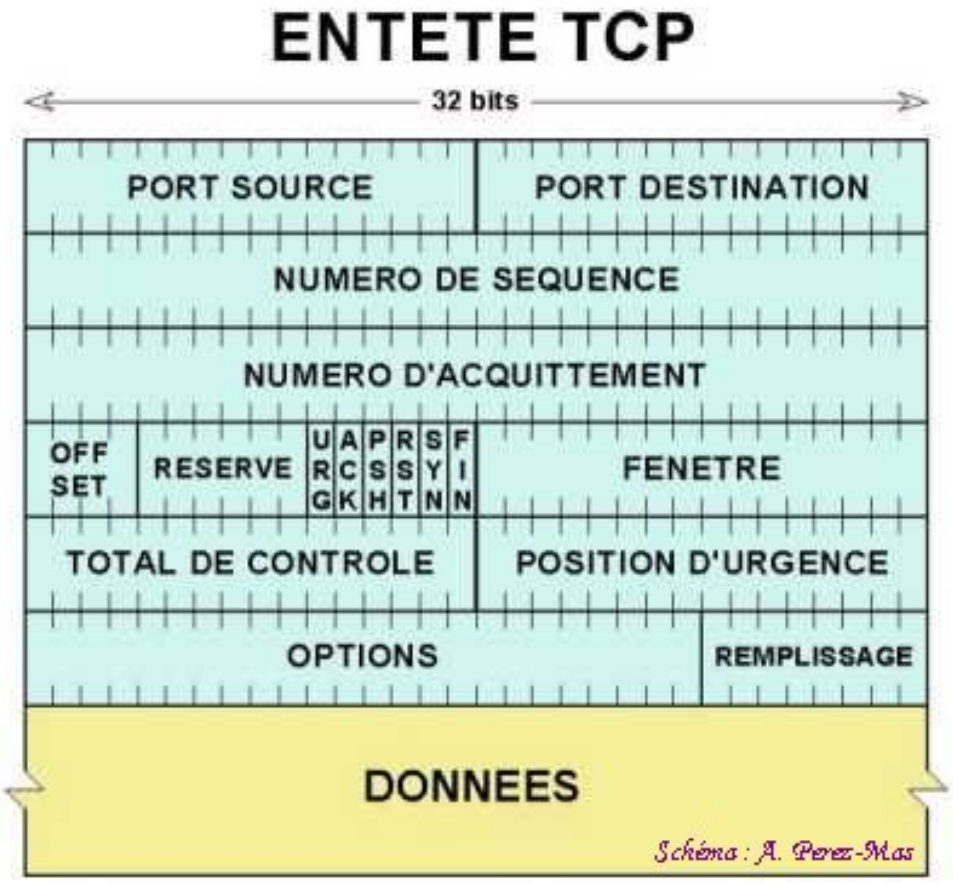

FIGURE 5 – Détail de l'en-tête TCP
Source: http ://arsene.perez-mas.pagesperso-orange.fr/reseaux/tcpip/tcp.htm

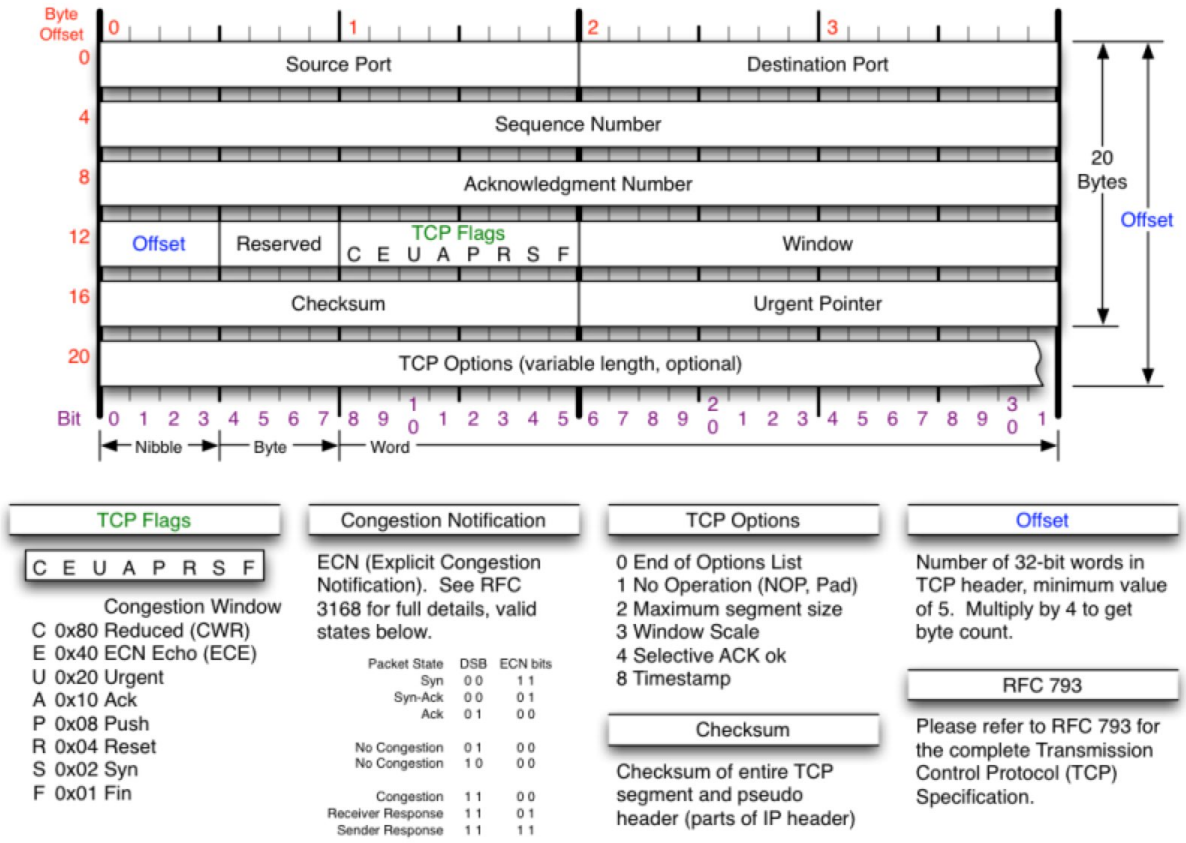

FIGURE 6 – Détail de l'en-tête TCP revisité : 3 nouveaux drapeaux
Source: https ://nmap.org/book/tcpip-ref.html

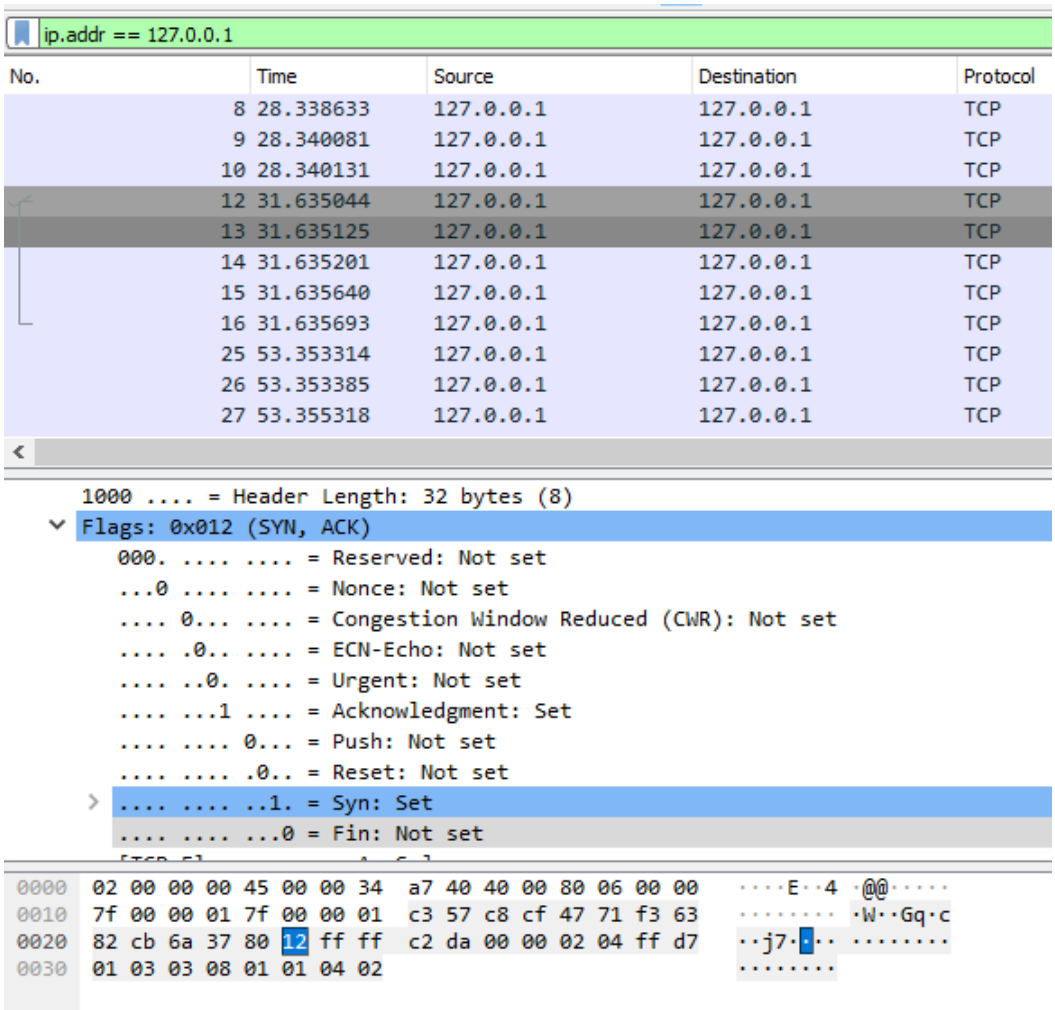

FIGURE 7 – Capture de la séquence d'initialisation d'une connexion client-serveur Python avec Wireshark

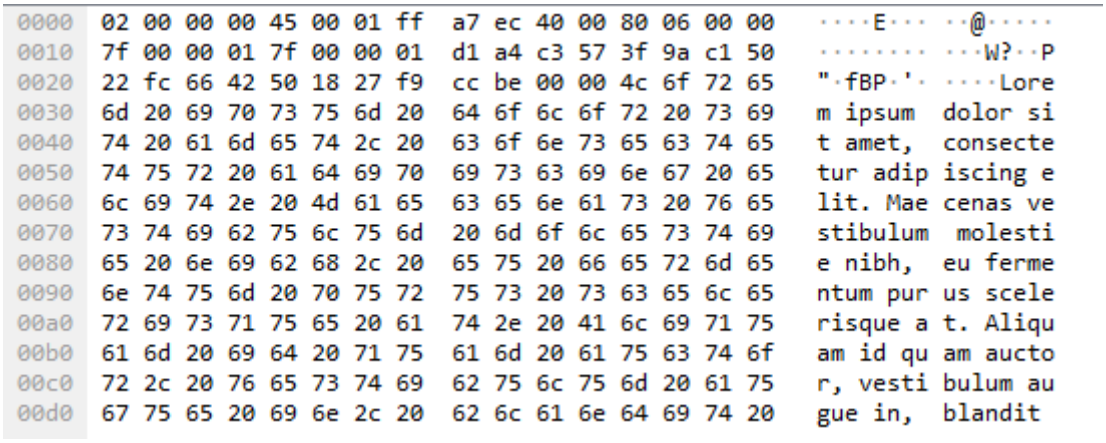

FIGURE 8 – Paragraphe de Lorem Ipsum intercepté et parfaitement lisible.