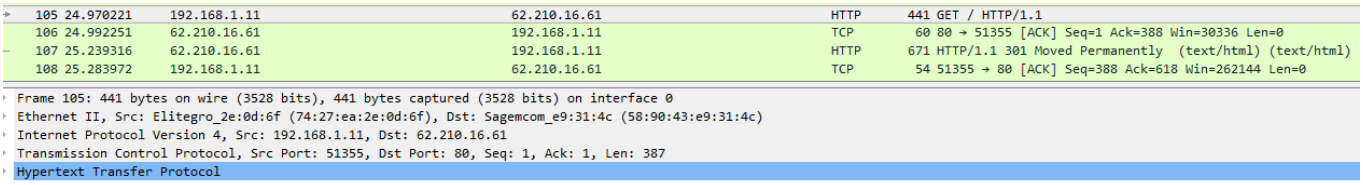
Transmissions de données

On va regarder comment se déroule une transmission de données sur un réseau et préciser le fonctionnement des protocoles TCP et IP. On utilise le logiciel Wireshark, qui permet d'écouter ce qui se passe sur le réseau.

1. Analyse d'une transmission de données

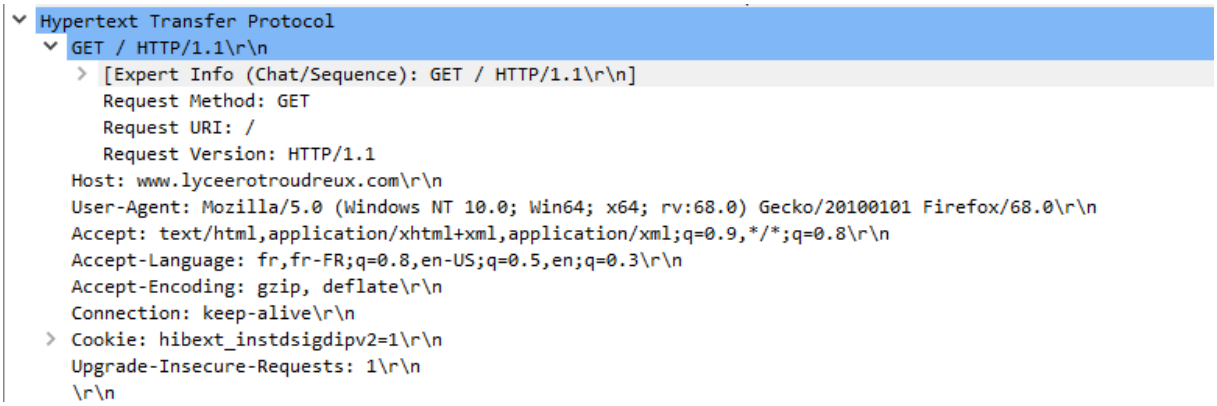
On tente de se connecter au site du lycée Rotrou, www.lyceerotroudreux.com.



Couche application

Le Protocole http va être sollicité.

Voyons en détail .



La méthode utilisée est la méthode GET et le site demandé est celui du lycée Rotrou.

Couche transport

Le protocole utilisé est TCP

```
Transmission Control Protocol, Src Port: 51355, Dst Port: 80, Seq: 1, Ack: 1, Len: 387
    Source Port: 51355
    Destination Port: 80
    [Stream index: 2]
    [TCP Segment Len: 387]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 388    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
```

Ici, les informations importantes concernent les ports utilisés. On retrouve notamment le port 80, réservé au protocole HTTP.

Couche réseau

```
Internet Protocol Version 4, Src: 192.168.1.11, Dst: 62.210.16.61
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 427
    Identification: 0x99a6 (39334)
  > Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x4ee4 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.1.11
    Destination: 62.210.16.61
```

Ici le protocole IP ajoute les adresses IP des deux ordinateurs qui doivent être mis en relation. On notera qu'à ce niveau, on sait que c'est le protocole TCP qui est utilisé (encapsulation)

Rappel : Chaque site est hébergé sur un serveur. Ce serveur doit donc être identifié de manière unique. Lorsque l'on tape www.lyceerotroudreux.com, on désire se connecter au serveur web (www) hébergeant le site du lycée. Pour qu'il n'y ait pas de confusion, comme pour le téléphone, chaque serveur possède un unique identifiant. C'est un nombre sous la forme w.x.y.z où w,x,y,z sont des entiers compris entre 0 et 255. Cet identifiant est appelé adresse IP (IPv4). Le lien entre adresse IP et nom de domaine est réalisée par le protocole DNS.

Il y a un autre type d'adresse IP (Ipv6) qui utilise la notation hexadécimale (c'est-à-dire à l'aide des symboles 0,1,2,3,4,5,6,7,8,9,A,B ,C,D ,E,F). Elle contient 32 symboles avec des séparations par « : » tous les 8 symboles.

Ex : 2019:0db8:0000:85a3:0000:0000:ac1f:8001.

Ici , l'adresse IP qui demande à se connecter au site du lycée est une adresse privée, non routable . Il s'agît de l'adresse IP d'un ordinateur sur un réseau local et qui passe par exemple par une box pour se connecter à internet.

Couche Accès Réseau

```
Ethernet II, Src: Elitegro_2e:0d:6f (74:27:ea:2e:0d:6f), Dst: Sagemcom_e9:31:4c (58:90:43:e9:31:4c)
  Destination: Sagemcom_e9:31:4c (58:90:43:e9:31:4c)
      Address: Sagemcom_e9:31:4c (58:90:43:e9:31:4c)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  Source: Elitegro_2e:0d:6f (74:27:ea:2e:0d:6f)
      Address: Elitegro_2e:0d:6f (74:27:ea:2e:0d:6f)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
```

Ici, le protocole Ethernet utilise les adresses MAC des cartes réseaux de deux ordinateurs devant communiquer.

On notera que le protocole ARP sert à faire le lien entre une adresse IP et une adresse MAC.

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Elitegro_2e:0d:6f (74:27:ea:2e:0d:6f)
    Sender IP address: 192.168.1.11
    Target MAC address: Sagemcom_e9:31:4c (58:90:43:e9:31:4c)
    Target IP address: 192.168.1.1
```

II. Protocole TCP

Le protocole TCP segmente les données à envoyer en paquets. Il est fiable et sans perte , contrairement au protocole UDP par exemple qui privilégie la vitesse de transmission plutôt que la fiabilité (UDP sera utilisé notamment pour le streaming ou les video conférences).

Une session TCP se présente de la sorte :

- l'établissement de la connexion
- les transferts de données
- la fin de la connexion.

Les données sont de ce type :

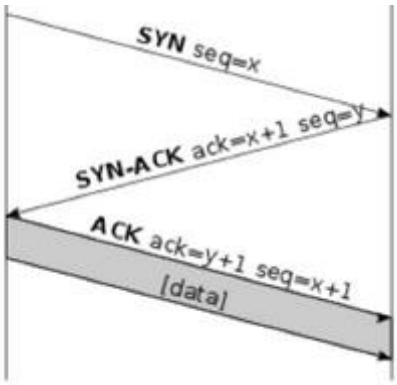
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Port Source 2 octets																Port destination 2 octets															
Numéro de séquence																															
Numéro d'acquittement																															
Taille de l'en-tête				Réservé				ECN		URG		ACK		PSH		RST		SYN		FIN		Fenêtre									
Somme de contrôle																	Pointeur de données urgentes														
Options																												Remplissage			
Données																															

(source wikipedia).

. Ports : Indique les ports sources et destinations (source 16 octets)

. Numéro de séquence : Il donne la position du segment dans le flux envoyé. Cela permettra au récepteur de remettre les segments dans l'ordre avant la transmission à la couche application.

. Numéro d'acquittement : il représente le numéro de séquence du destinataire.



Syn : demande de connexion, Ack : acquittement.

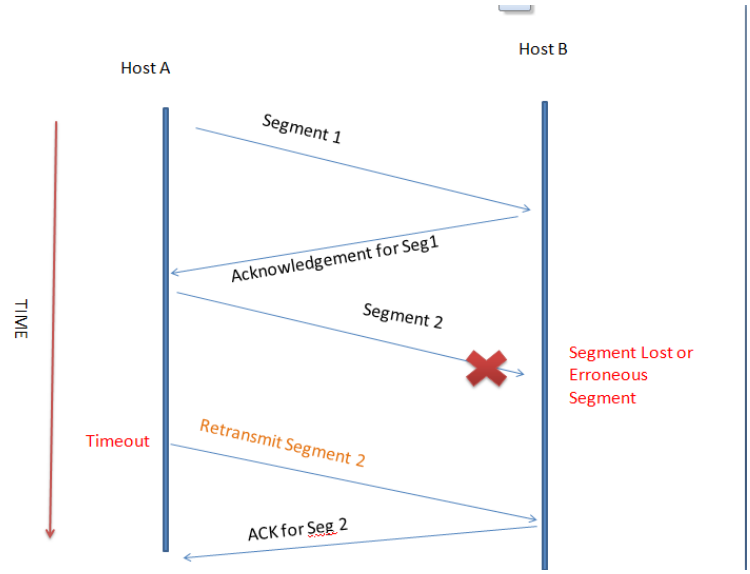
Maintenant que la connexion est établie, comment assurer le bon acheminement des données sans perte ?

L'émetteur est sûr d'envoyer tous les segments mais n'est pas sur qu'ils soient tous reçus.

Un système de temporisation est mis en place : On renvoie un segment si l'on n'a pas reçu d'acquittement au bout d'un certain temps. Ce temps n'est pas choisi au hasard. Il ne le faut pas trop court, pour éviter d'envoyer des segments déjà reçus et pas trop long. Le délai mis en place est légèrement supérieur à la durée que met un segment pour effectuer l'aller-retour entre les deux machines.

Est on sauvé ? Ben non. En effet, le message peut bien être arrivé , mais l'acquittement s'est perdu . Ou le message s'est perdu, cas moins grave, il suffit de le renvoyer. Dans le premier cas, on peut renvoyer le message à tort.

Exemple :



Ici, pas de problème pour le segment 1. Le segment 2 n'arrive pas, une fois le délai de temps dépassé, a envoie le segment. Dans cet exemple, pas de souci. Mais il se pourrait que ce soit l'acquittement qui se soit perdu.

Pour régler le problème, on va ajouter un nombre au données envoyées. Il n'y a besoin que deux nombres différents, 0 et 1.

En effet, on commence le compte à 0. Le premier message arrive. L'émetteur attend un acquittement avec le chiffre 0 . S'il le reçoit à temps, il envoie le second message et attend un acquittement avec 1. Puis le troisième avec un acquittement comportant un 0. Etc. C'est le protocole du bit alterné. Cet ajout se fait au niveau de la couche accès réseau, donc dans une trame.

Supposons que l'émetteur ne reçoive pas l'acquittement pour son premier message, qui est bien reçu cependant. Il le renvoie. Cependant le récepteur attend un message avec le chiffre 1. Il renvoie alors un acquittement avec le chiffre 0, indiquant à l'émetteur qu'il attend le deuxième message et qu'il a bien reçu le premier. Ainsi, on est sur que tous les messages vont bien arriver.

Bien évidemment, il y a une perte de temps importante avec ce système, qui sert de base à des systèmes plus performants comme le *sliding window* qui permet à l'émetteur d'envoyer plusieurs messages avant de recevoir les acquittements.

A retenir :

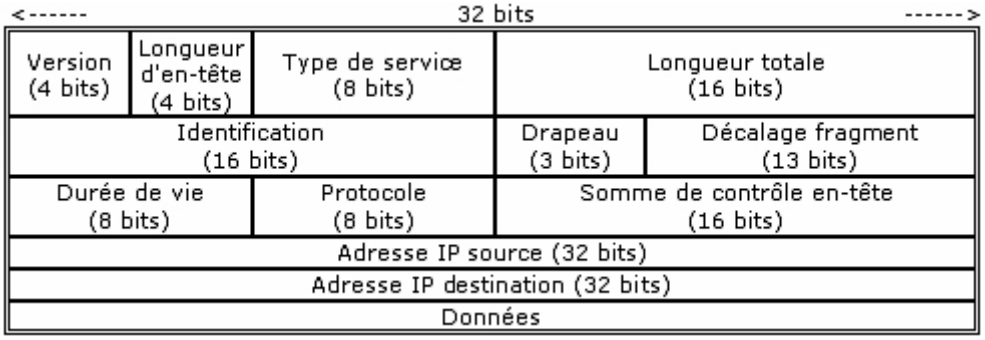
La fiabilisation de la transmission TCP est assuré par :

- L'envoie d'un accusé de réception à chaque messages reçus. (Acquittement)
- L'instauration d'un délai au-delà duquel on renvoie le message si l'on n'a pas reçu l'acquittement
- Une numérotation basique permettant de vérifier si tos les messages ont bien été reçus.

III . Protocole IP

Le protocole IP se charge de l'acheminement des paquets. Il ne s'occupe pas des données, de l'ordre d'envoi…il joue en fait le rôle d'un service de courrier. Il connaît l'adresse de l'expéditeur et celle du récepteur.

Ce protocole de plus bas niveau que TCP l'encapsule donc et ajoute principalement les adresses IP des deux communicants.



Le protocole IP est un protocole non connecté, c'est-à-dire qu'il assure une transmission de données dans laquelle chaque paquet est préfixé par un entête contenant une adresse de destination, suffisante pour permettre la livraison autonome du paquet, sans recours à d'autres instructions. (wikipedia)