Tohotcheu Reine

TP2 – Réseau

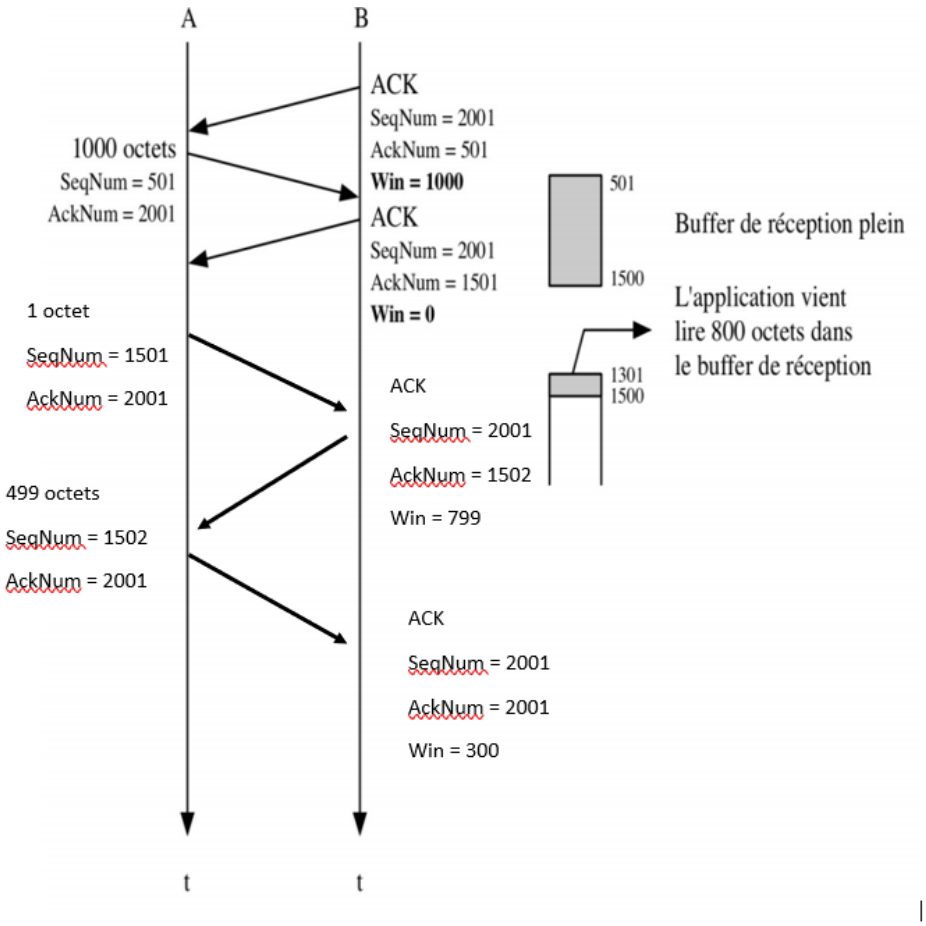
1. Contrôle de flux dans TCP

Question 1.1

Après l'envoi du segment TCP avec SeqNum = 1227, AckNum = 2513 et Win = 3007, la plage que l'on peut accepter est la valeur d'AckNum plus la valeur de la taille de la fenêtre Windows.

En fait, la plage est la valeur d'AckNum qui est renvoyé et qui peut être acceptée. Lorsque l'ack nous montre qu'il peut accepter 3007, on sait que pour avoir la plage totale, on calcul 2513+3007 = 5520. Donc, la plage d'octet que la machine est prête à recevoir est [2513 ; 5520].

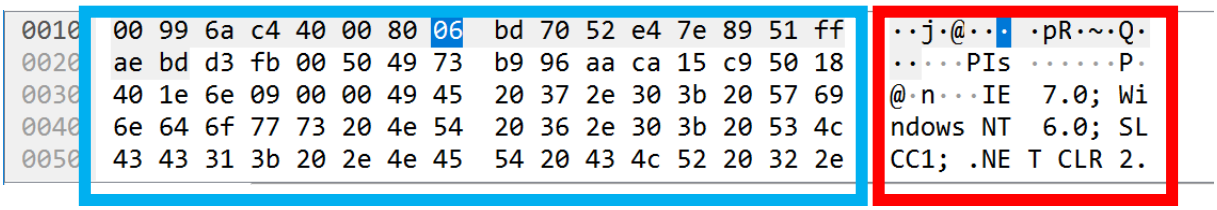
Question 1.2



2. Continuation sur WireShark

Question 2.2

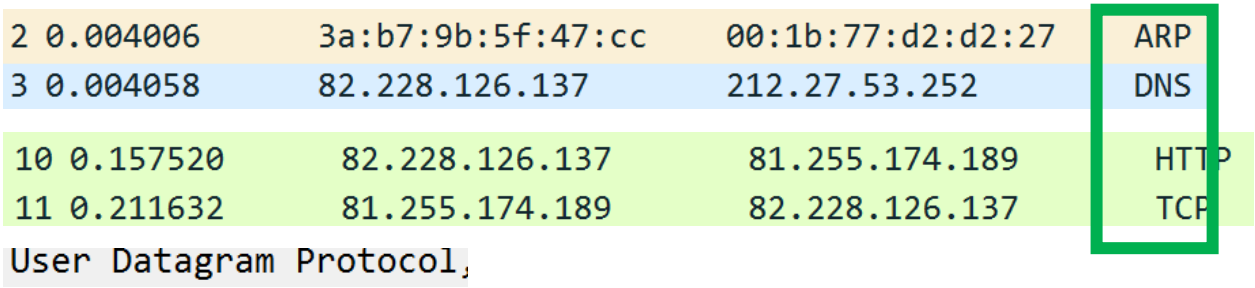
Les données sont représentées en formats hexadécimal (encadré en bleu) et ASCII (encadré en rouge) :



Question 2.3

Sur la capture affichée nous pouvons observer les protocoles cinq protocoles différents.

Les protocoles : TCP, HTPP, DNS, UDP et ARP (ils sont visibles sur la capture d'écran ci-dessous, extraite de WireShark) :



Le protocole TCP est un protocole de transport aussi appelé modèle TCP/IP. Dans le modèle OSI, il correspond à la couche transport, intermédiaire de la couche réseau et de la couche session.

Le protocole HTTP est un protocole de la couche application. Il utilise le protocole TCP comme protocole de transport.

Le protocole DNS sert à traduire les noms de domaine Internet en adresse IP ou autres enregistrements.

UDP est un protocole de transport, on voit qu'il est utilisé comme sous protocole lorsque le transport de données DNS.

Le protocole ARP sert à lier les adresses MAC aux adresses IP.

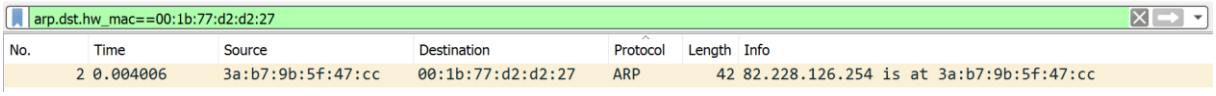
Question 2.4

La version WireShark que nous utilisons est capable d'analyser 3000 de protocoles selon les informations extraites du site.

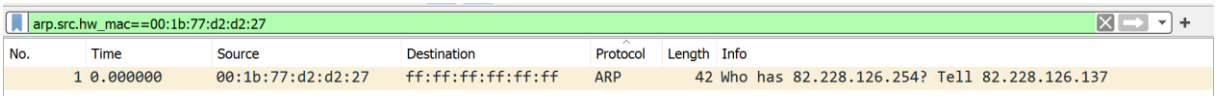
3. Filtres d'affichage WireShark

Avec les filtres que nous avons appliqués, nous voyons bien que les filtres ne sélectionnent que les trames ARP ayant l'adresse MAC 00:1b:77:d2:d2:27.

Dans le premier cas, nous mettons un place un filtre qui grâce à l'adresse MAC nous permet d'obtenir la trame avec l'adresse de destination ARP concernée : arp.dst.hw_mac==00:1b:77:d2:d2:27



Avec la formule suivante, arp.src.hw_mac==00:1b:77:d2:d2:27, nous avons l'adresse source :



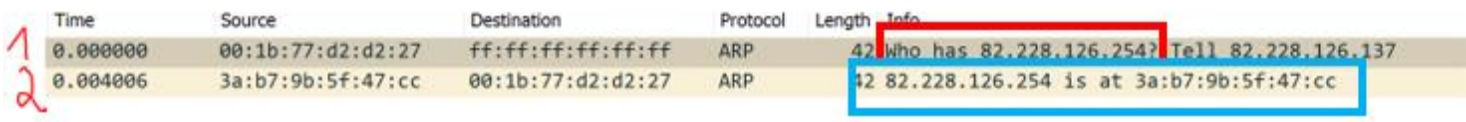
Ce filtre nous permet donc d'avoir les trames avec les adresses de sources et destinations avec l'adresse MAC de l'énoncé.

4. Trace HTTP

Question 4.1

On sait que le rôle d'ARP est d'associer les adresses MAC aux adresses IP, sur la capture nous voyons que lors du premier échange le protocole ARP demande qui a l'adresse « 82.228.126.254 » (encadré en rouge sur la capture d'écran) en mode broadcaste c'est-à-dire qu'il va l'envoyer à toutes les machines d'où l'adresse de destination « ff :ff :ff :ff :ff :ff » de la trame 1.

Ensuite, la machine concernée va répondre et fournir l'adresse IP associé à l'adresse MAC (encadrée en bleu) :



Ainsi, les destinataires de la première trame sont toutes les machines impliquées dans le réseau, car l'envoi est un broadcaste.

Les adresses participants à cette échanges sont : *3a :b7 :9b :5f :4f :cc* et *99 :1b :77 :d2 :d2 :27.*

Question 4.2

L'objectif des trames 3 à 6 est d'accéder au serveur du site ratp.fr afin d'accéder au site.

Le site visé est donc ratp.fr :

Si nous observons de détail de manière plus précises, et que nous cliquons que la trame 3, nous voyons que la requête envoyée est de type A :

```
ratp.fr: type A, class IN
   Name: ratp.fr
   [Name Length: 7]
   [Label Count: 2]
   Type: A (Host Address) (1)
   Class: IN (0x0001)
```

C'est une requête IPV4, dans cette trame on voit qu'il y a une demande pour obtenir l'adresse du serveur ratp.fr, la réponse est au niveau de la trame 4. On remarque qu'il y a deux adresses, car il y a peut-être deux serveurs associés au site ratp.fr donc le DNS a traduit les 2 adresses suivante :

```
Answers
   ratp.fr: type A, class IN, addr 81.255.174.189
   ratp.fr: type A, class IN, addr 62.160.111.1
```

Ensuite au niveau de la trame 5, on voit qu'il y a une nouvelle demande de type AAAA, avec l'IPV6 :

```
Queries
   ratp.fr: type AAAA, class IN
      Name: ratp.fr
      [Name Length: 7]
      [Label Count: 2]
      Type: AAAA (IPv6 Address) (28)
      Class: IN (0x0001)
   [Response In: 6]
```

La réponse à la trame 6 ne contient pas d'adresse IP :

```
Authoritative nameservers
  ratp.fr: type SOA, class IN, mname ns0.ratp.fr
      Name: ratp.fr
      Type: SOA (Start Of a zone of Authority) (6)
      Class: IN (0x0001)
      Time to live: 151 (2 minutes, 31 seconds)
      Data length: 39
      Primary name server: ns0.ratp.fr
      Responsible authority's mailbox: hostmaster.ratp.fr
      Serial Number: 2007112204
      Refresh Interval: 21600 (6 hours)
      Retry Interval: 3600 (1 hour)
      Expire limit: 604800 (7 days)
      Minimum TTL: 300 (5 minutes)
```

On suppose qu'il n'a peut-être pas trouvé l'adresse IP associée cependant la réponse transmet de nombreuses informations sur la zone DNS dont le site primaire ns0.ratp.fr ainsi que les routes de back up.

Question 4.3

Le protocole de transport associé au protocole applicatif est l'UDP (User Datagram Protocol).

```
> User Datagram Protocol, Src Port: 51036, Dst Port: 53
```

Grace à se protocole de transport, les données vont transiter entre les serveurs.

Ce protocole ne fournit pas de contrôle d'erreur il n'est pas orienté connexion de la couche transport. Il permet de transmettre rapidement des petites quantités de données depuis un serveur vers de nombreux clients. Ici, le protocole est adapté car les données échangées ne sont pas nombreuses.

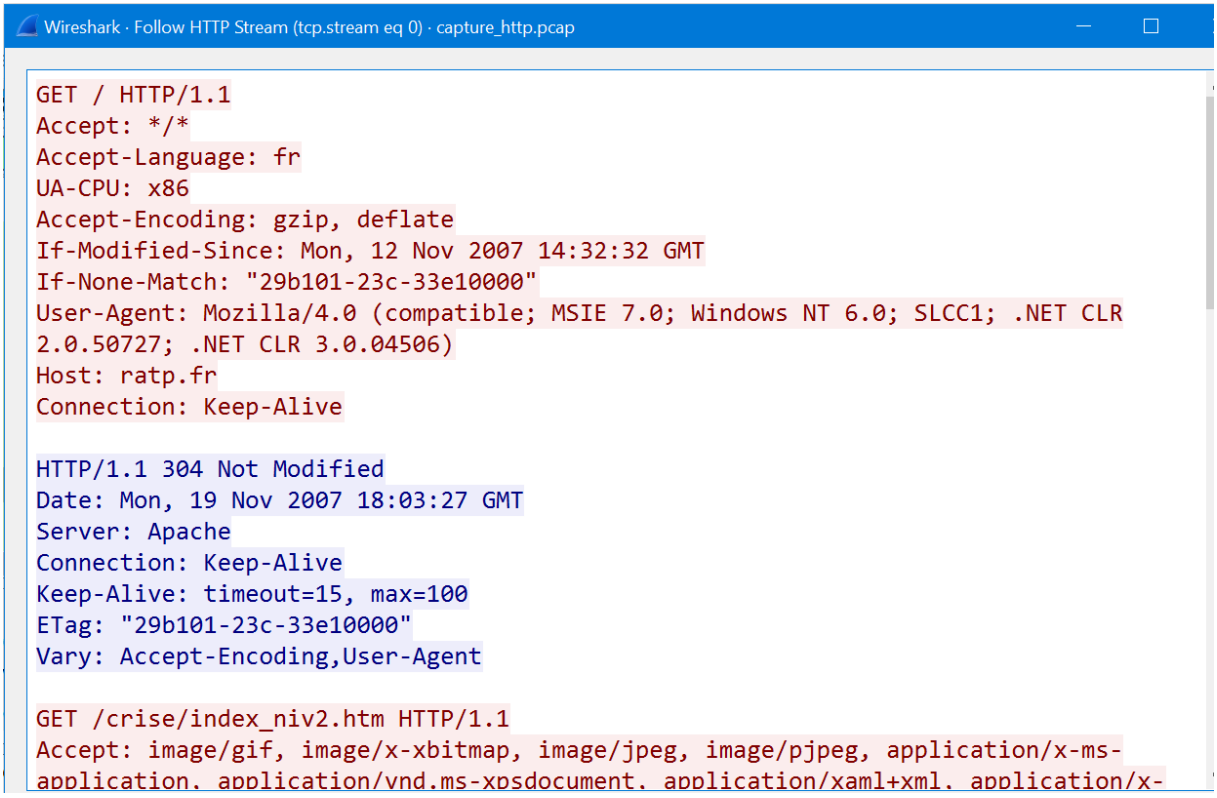
Question 4.4

Le but de cet échange est d'obtenir l'adresse IP du serveur associé au site ratp.fr. (Cf. Question 4.2)

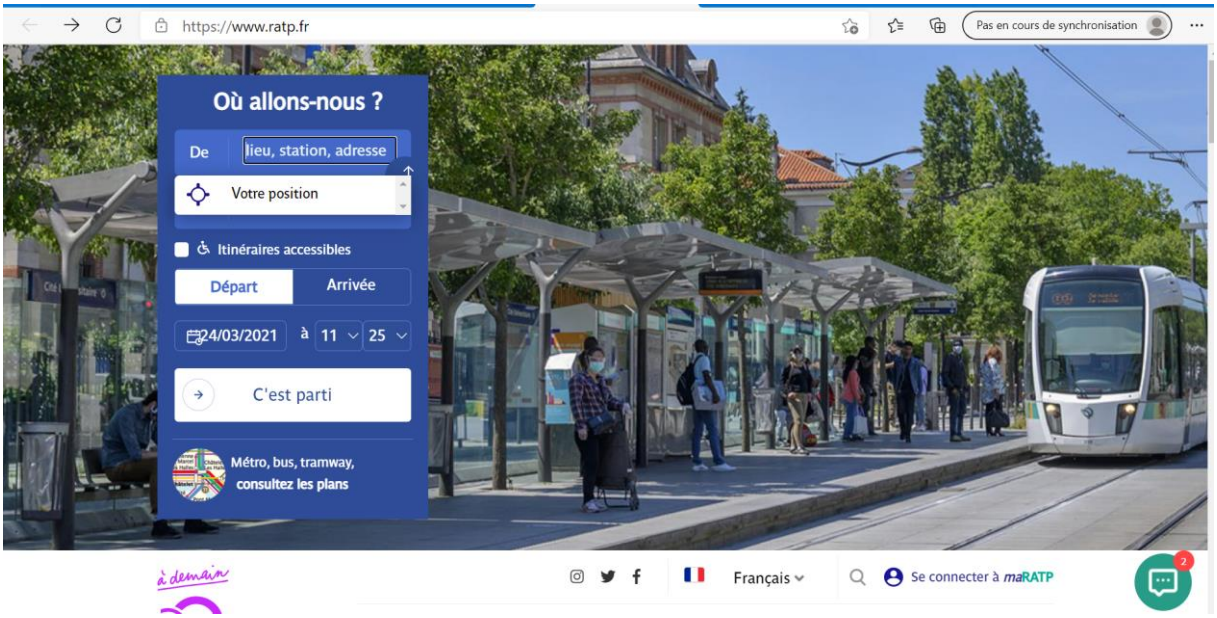
```
G>..........ratp.fr.....G>..........ratp.fr..............'.ns0..
hostmaster..w.....T`.....          :....,
```

Question 4.5

Si on clique sur Follow → http Stream, on obtient ceci :

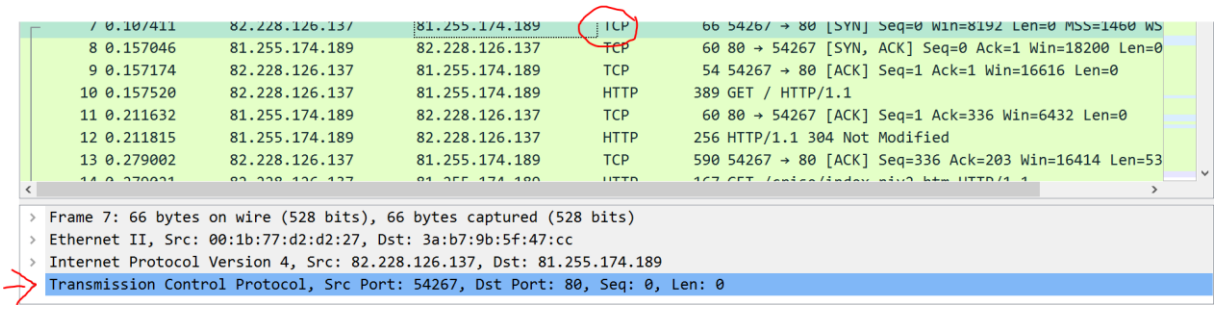


On voit que les GET sont des requêtes qui permettent de demander une ressource, par exemple un fichier HTML, au serveur Web. Via le site ratp.fr nous avons plusieurs onglets liés à celui si donc plusieurs GET sont nécessaires pour accéder aux différentes pages :



Question 4.6

Le protocole utilisé à partie de la trame 7 est le protocole TCP :



Grace à ce protocole de transport nous pouvons nous pouvons transporter plus de données, de manière sécurisée et fiable.

Remarques et différences entre les en-têtes TCP et UDP.

On remarque que lorsque le protocole de transport est différents l'en-tête change.

En effet, le protocole UDP contient moins d'informations dans son en-tête :

```
User Datagram Protocol, Src Port: 53, Dst Port: 51036
    Source Port: 53
    Destination Port: 51036
    Length: 84
    Checksum: 0xab04 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
```

Il n'y a pas de sécurité à proprement parlé (avec Ack et autre), les données qui transitent par ce protocole sont peu nombreuse.

Alors que pour le protocole TCP on voit qu'il est sécurisé notamment avec la présence de l'Ack dans son en-tête, ce protocole assure la réception des données.

```
Transmission Control Protocol, Src Port: 54267, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 54267
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 1232320046
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    1000 .... = Header Length: 32 bytes (8)
```

Les données échangées sont plus nombreuses, il y a des échanges de plusieurs segments contrairement à l'UDP qui permet des échanges minimes avec un seul segment.

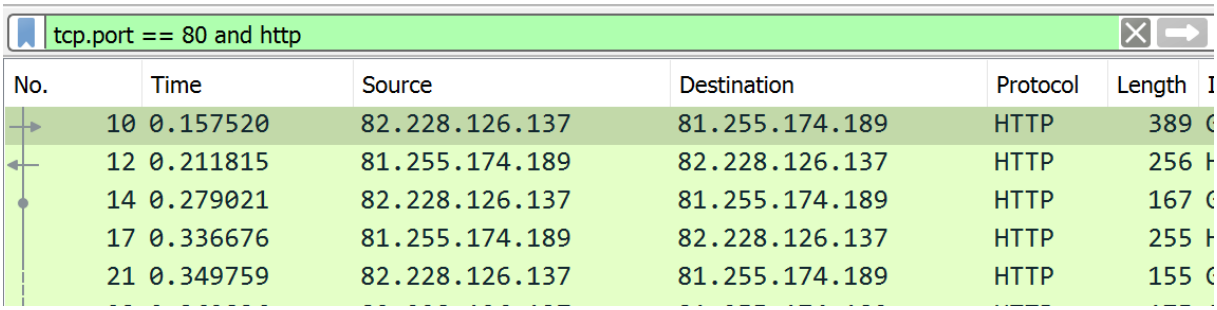
Question 4.7

Grace à l'énoncé on voit que pour filtrer les trames correspondant au protocole TCP il faut appliquer ce filtre :

```
tcp.port == 80
```

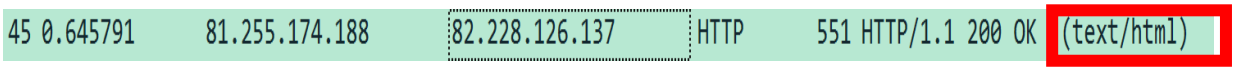
Donc pour notre filtre permettant d'obtenir les trames relatives au protocole http et utilisant le port 80, avec plus de détails on utilise : tcp.port == 80 and http.

Ainsi le filtre permet de répertorier toutes les trames qui utilisent le port 80 et qui ont comme protocole de transport http.



Question 4.8

La trame 45 correspond à du texte/HTML, soit le contenu du fichier HTML lié au site :



On retrouve les détails de la page HTML visée en cliquant sur *Follow* → *HTTP Stream* :

Nous voyons que l'on a accès à la rubrique « Information trafic RATP » :

```
<title>Information trafic RATP</title>
```

Nous avons aussi de nombreuses informations dans *HTTP Stream*, comme l'url des logos utilisés : `background: url(/picts/v_home_ratp/niv3bis/logo_ratp.gif) no-repeat;`,

Le corp (body en HTLM) correspond au contenu principal de la page web, ici nous avons le contenu de la rubrique informations trafic du site ratp.fr :

```
/* */
body
{
/*text-align:center;*/
margin-left:0;
margin-top:0;
}
#page{
/*width:800px;
margin-left:auto;
margin-right:auto;*/
text-align:left;
}
```

On retrouve aussi les titres (header du site), les en-têtes h1, h2, h3 et h4, dont h2 qui contient l'url d'un gif :

```
h2
{
position:relative;
background: url(/picts/v_home_ratp/niv3bis/exclam.gif) no-repeat;
```

.

Ensuite nous retrouvons les renseignements importants :

```
#renseignements
{
display:block;
float:left !important;
```

, les liens annexes et les infos trafic sur les lignes de métros par exemple :

```
<br><a href="trafic.php?cat=1" class="normal">M.tro</a>
<br>Ligne 1 : 1 rame toutes les 5 min
<br>Ligne 2 : 1 rame toutes les 10 min
<br>Lignes 3 et 3bis : 1 rame toutes 25 min
<br>Ligne 4 : 1 rame toutes les 10 min
<br>Ligne 5 : 1 rame toutes les 20 min
<br>Lignes 6, 9, 10 et 12 : service quasi nul
<br>Ligne 7 : 1 rame toutes les 40 min
<br>Ligne 7bis : 1 rame toutes 20 min
<br>Ligne 8 : 1 rame toutes les 25 min
<br>Ligne 11 : 1 rame toutes les 15 min
<br>Ligne 13 : 1 rame toutes les 10 min (Branche G.P.ri - Asni.res
Gennevilliers non assur.e)
<br>Ligne 14 : fonctionne normalement
<br>Orlyval : trafic normal (RER B non assur. . la Gare d'Antony)
```

Il y aussi des informations sur les tramways et les RER, on remarque aussi que le langage de script utilisé est le JavaScript1.1 : `<script language="JavaScript1.1">`.

La capture_http.pcap WireShark sur laquelle nous travaillons date de 2007 : `mardi 20 novembre 2007`.

Donc, l'interface à changer depuis et nous ne pouvons pas forcement vérifier tous les aspects présents sur http Stream.

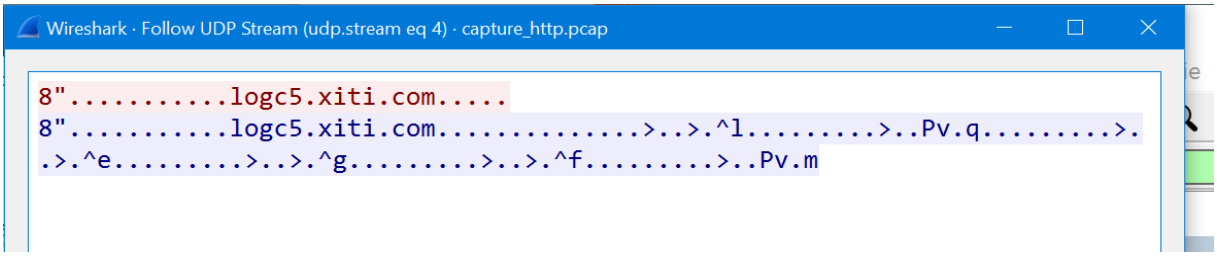
Le fichier correspondant a été envoyé en plusieurs fois, nous voyons qu'il y a plusieurs trames :

```
[Request in frame: 37]
[Next request in frame: 46]
[Next response in frame: 49]
[Request URI: http://www.ratp.info/picts/v_home_ratp/niv3bis/reseaux.gif]
```

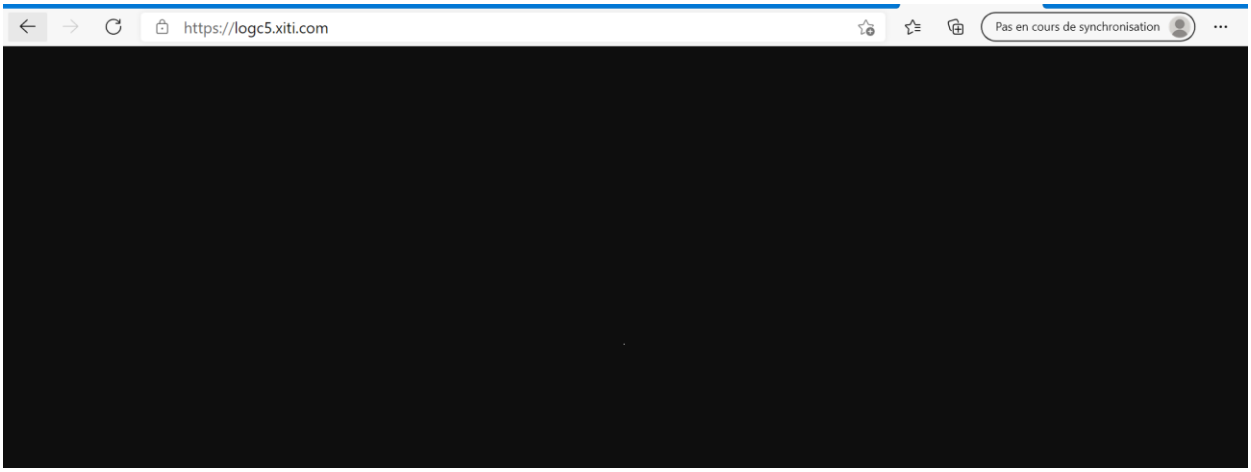
Question 4.9

Lorsqu'il y a un autre nom de domaine, nous savons que nous sommes face à un nouveau site donc une nouvelle requête DNS est faite pour accéder au site.

Ainsi, la seconde requête DNS de la trame 48 permet d'accéder au site « logc5.xiti.com » :



Lorsque l'on accède à se site via un navigateur, on tombe sur cette fenêtre noire avec un point blanc au milieu :



Ce site permet surement de calculer le flux ou encore de contrôler le trafic. On peut aussi imaginer que c'est un pop-up ou même une page qui permet de répertorier l'affluence des utilisateurs qui visitent ratp.fr.

Le site de la RATP utilise des « sous sites » qui ont différentes fonctions, d'où la requête DNS (trame 48) qui permet de faire des calculs ou répertorier des informations annexes n'ayant pas d'impact direct sur l'échange, nous avons juste une Query DNS pour accéder à l'autre site. Donc, cette requête n'est pas liée au reste de l'échange.