## Exercice 1

Deux clients C1 et C2 commencent en même temps une session http avec un serveur S.

— C1 envoie une requête vers S. Quel est le numéro de port source et le numéro de port destination ?

Port source : numéro de port X aléatoire supérieur à 1024

Port destination : 80 (port d’écoute serveur web) (si http**S** : 443)

— C2 envoie une requête vers S. Quel est le numéro de port source et le numéro de port destination ?

Port source : numéro de port Y aléatoire supérieur à 1024

Port destination : 80 (port d’écoute serveur web) (si http**S** : 443)

— S envoie une réponse vers C1. Quel est le numéro de port source et le numéro de port destination ?

Port source : 80 (port de données serveur web) (si http**S** : 443)

Port destination : numéro de port X (aléatoire supérieur à 1024)

— S envoie une réponse vers C2. Quel est le numéro de port source et le numéro de port destination ?

Port source : 80 (port de données serveur web) (si http**S** : 443)

Port destination : numéro de port Y (aléatoire supérieur à 1024)

## Exercice 2

Vrai-Faux

— Le protocole HTTP est un protocole sans état.

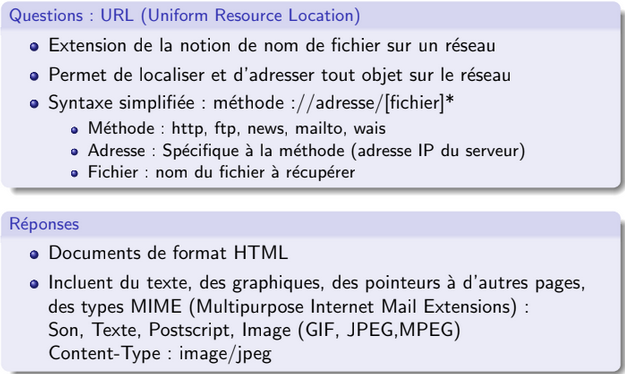
**Vrai** (page 70) (<https://fr.wikipedia.org/wiki/Protocole_sans_%C3%A9tat>)

(Chaque requête est indépendante)

— Un utilisateur demande une page Web qui contient du texte et deux images. Pour cette page, le client enverra une seule requête et recevra trois réponses.

**Faux** : il y a autant de requêtes que d'objets à télécharger sur la page. (ici 3 objets 3 requêtes)





— Deux pages web distinctes http://www.loria.fr/la-recherche et http://www.loria.fr/les-actus peuvent être envoyées sur la même connexion persistante.

**Vrai**, (page 70) à partir de HTTP 1.1, ouverture d’une connexion TCP avec [www.loria.fr](http://www.loria.fr) et on émet plusieurs requêtes dans cette même connexion TCP.

## Exercice 3

Depuis votre navigateur, vous **entrez une URL** pour obtenir une page web. **L’adresse IP associée à l’URL n’est pas cachée** localement sur votre machine.

Supposons que ***n*** serveurs DNS sont interrogés afin d’obtenir l’adresse IP ; ces visites représentent des délais aller-retour de RTT1 pour le premier serveur DNS, . . ., RTTn pour le ***n*** eme serveur DNS.

Supposons également que la page web demandée consiste seulement d’un texte HTML. Soit RTT0, le RTT entre votre machine et le serveur web contenant la page demandée.

- En négligeant le temps de transmission de la page (RTT0), déterminer en nombre de RTT, le temps écoulé entre le moment où vous cliquez sur le lien et celui où vous recevez la page web.

Le temps total est : (temps pour obtenir l’adresse IP) + (connexion TCP avec le serveur web) + (temps pour envoyer la requête GET et recevoir la page HTML)

- La page HTML contient maintenant 8 objets (images,...) sur le même serveur. Toujours en négligeant le temps de transmission, déterminer en nombre de RTT, le temps écoulé entre le moment où vous cliquez sur le lien et celui où vous recevez la page web complète avec tous les objets dans les cas suivants :

— connexion HTTP non-persistante :

Temps pour obtenir l’adresse IP :

Temps pour établir une connexion TCP :

Temps pour envoyer une requête GET et recevoir la page html :

Pour chaque objet établir TCP et GET :

— connexion HTTP non-persistante avec le navigateur configuré avec 5 connexions TCP en parallèle :

Temps pour obtenir l’adresse IP :

Temps pour établir une connexion TCP :

Temps pour envoyer une requête GET et recevoir la page html :

Pour chaque objet établir TCP et GET (5 connexions en parallèle) :

(5 objets (2RTT0) + 3 objets (2RTT0))

Total :

— connexion HTTP persistante sans pipelining. :

Temps pour obtenir l’adresse IP :

Temps pour établir une connexion TCP :

Temps pour envoyer une requête GET et recevoir la page html :

Temps pour récupérer les 8 objets (persistant) :

Total :

— connexion HTTP persistante avec pipelining. :

Temps pour obtenir l’adresse IP :

Temps pour établir une connexion TCP :

Temps pour envoyer une requête GET et recevoir la page html :

Temps pour récupérer les 8 objets (1 seul car on récupère tout en même temps) :

Total :

## Exercice 4

1. Soit un utilisateur voulant à partir d’un navigateur web accéder au site www.univ-lorraine.fr, quels protocoles au niveau transport et application seront utilisés ?

Protocole de transport : TCP(connexion HTTP), UDP (Pour faire la résolution DNS)

(Autre protocole hors exo : SCTP, SPX, ATP)

Protocole application : HTTP, HTTPS, Gopher, SMTP, SNMP, FTP, Telnet, NFS, NNTP



2. Soit la chaîne de caractères ASCII capturée par wireshark suite à une action effectuée depuis un navigateur web :

GET /content/faire-dialoguer-les-savoirs-cest-innover HTTP/1.1

Host: www.univ-lorraine.fr

User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:42.0) Gecko/20100101 Firefox/42.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8 Accept-Language: fr,fr-FR;q=0.8,en-US;q=0.5,en;q=0.3

Accept-Encoding: gzip, deflate

Connection: keep-alive

A partir de ces informations, répondez aux questions ci-dessous :

— Quelle est l’URL du document qui a été saisie dans le navigateur ?

URL : http://www.univ-lorraine.fr/content/faire-dialoguer-les-savoirs-cest-innover

* protocole d’accès
* l’hôte
* adresse/chemin du contenu

— Quelle est la version HTTP utilisée par le navigateur ?

Version : HTTP 1.1

— Est-ce que le navigateur demande une connexion persistante ou non ? Justifiez votre réponse.

Connexion persistance car “Connection : **keep-alive**” (rester en vie)

— Connaît-on l’adresse IP du serveur qui est contacté ?

Non on ne connaît que son nom elle n'apparaît pas (pas directement) (le DNS n’a pas encore calculé l’adresse IP)

— Quel est le type de navigateur qui a généré la requête ?

Mozilla (depuis un MAC OS 10) Firefox (Il faut regarder le User-Agent)

## Exercice 5

- Quel protocole applicatif utilise le client ftp pour obtenir l’adresse IP correspondant à ftp.ripe.net ? Quel protocole de transport utilise ce protocole applicatif ?

Protocole applicatif : DNS

Protocole de transport : Par défaut UDP (on tend vers du DNS au dessus de HTTP donc TCP)

— Le client ftp se connecte ensuite sur le serveur ftp.ripe.net.

— Quel protocole applicatif est utilisé ?

FTP

— Quel protocole de transport est utilisé ?

TCP

— Quel est le numéro de port côté serveur ?

21

— Quel est le numéro de port côté client ?

Port aléatoire (au dessus de 1024)

— Lorsque l’utilisateur envoie son login et son mot de passe, quels sont les APDUs qui sont envoyées du client vers le serveur. Définir avant la notion de APDU.

Application Protocol Data Unit (APDU)

Lien pour le fonctionnement du protocole ftp : <https://tools.ietf.org/html/rfc959>

4.1.1. ACCESS CONTROL COMMANDS

USER et PASS

— Le client souhaite maintenant récupérer le contenu d’un fichier. Le mode de transfert de données peut être actif ou passif.

— Si le mode est actif, qui établit la connexion au niveau transport pour transférer les données ? Quel sera le numéro du port côté serveur ? côté client ?

Le serveur établit la connexion

— Si le mode est passif, qui établit la connexion au niveau transport pour transférer les données ? Quel sera le numéro du port côté serveur ? côté client ?

Le client établit la connexion vers le serveur

## Exercice 6

Quelle est la différence entre ftp et tftp ?

ftp : en TCP/IP

tftp : en UDP/IP

TFTP est au dessus de FTP au niveau de la couche transport