Faculté des Sciences & Techniques — Université de Limoges

Master 1ère année

Protocoles & Prog. Réseau

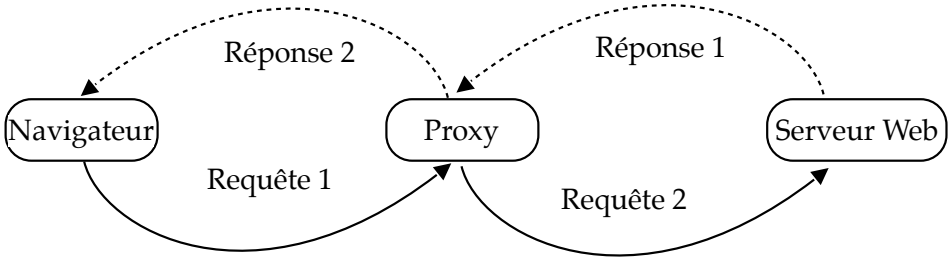
TP n°1

Réalisation d'un proxy pour le Web (protocole HTTP)

Un serveur « *Proxy* » est un processus « *mandataire* », qui va servir **d'intermédiaire** pour la communication entre le client Web (le navigateur) et le serveur Web.
Cette communication est organisée suivant le protocole HTTP, basé TCP.

Le rôle du proxy est de :

- **filtrer** les communications en interdisant l'accès à des sites en fonctions de leur contenu (suivant une liste de mots-clé) ;
- servir de « *cache* », c-à-d de **mémoriser** le contenu des pages le plus souvent visitées et éviter ainsi de devoir les récupérer de nouveau auprès du serveur qui les hébergent.
  Le but est de réduire le trafic de données en direction de l'extérieur (le proxy étant en général dans le réseau local).
- **modifier** les données transmises (réalisation de censure en vue d'une protection parentale par exemple) ;
- assurer une **meilleur sécurité**. Les machines clientes ne dialoguent pas directement avec les serveurs situés à l'extérieur du réseau local, mais uniquement avec le proxy (on peut également interdire toute transaction directe, sans passer par le proxy, d'un client vers un serveur extérieur à l'aide d'un *firewall*).



Fonctionnement du proxy

Le **navigateur** doit d'abord être configuré pour utiliser un proxy pour le protocole (ou service) HTTP :

- le proxy doit se comporter comme un **serveur** :
  - il attendra des connexions du navigateur ;
  - il doit disposer d'un SAP (adresse IP, n° de port) ;
  - ces informations sont à renseigner au niveau des préférences du navigateur.

Le proxy :

- crée une « *socket* » pour permettre au navigateur de se connecter à lui ;
- **reçoit une requête** de la part du navigateur (« *Requête 1* » sur la figure) ;
- **décompose** l'URL présente dans cette requête afin de récupérer :
  - le nom du serveur Web hébergeant la ressource ;
  - le numéro de port s'il est indiqué (par défaut c'est le port 80) ;
  - le chemin d'accés au document ;
- crée une **seconde** « *socket* » pour pouvoir se connecter au serveur Web ;
- **envoie une requête** au serveur Web en reprenant de manière légèrement modifiée la requête du navigateur (« *Requête 2* » sur la figure) ;
- **récupère les informations** en réponse du serveur Web (« *Réponse 1* » sur la figure) ;
- **renvoie ces informations** en réponse au navigateur Web (« *Réponse 2* » sur la figure).
- **attend une nouvelle connexion** du navigateur.

Rappel sur HTTP

Le navigateur peut transmettre des commandes « GET » ou « POST ».
Chacune de ces commandes termine par une ligne vide signifiant que le navigateur n'a plus rien à transmettre au serveur :

- commande « *GET* » : la plus courante pour réclamer une ressource à un serveur Web ;
- commande « *POST* » : elle permet de transmettre au serveur Web, en plus de la requête d'accès à une ressource, des données au format MIME :
  - transmises par le navigateur à la suite de sa commande « POST » après une ligne vide ;
  - accompagnées d'une taille, qui permet de les récupérer, indiquée par « `Content-Length: longueur_en_octets` ».

Format des requêtes du navigateur dans le cas de l'utilisation d'un proxy

Le navigateur envoie une requête « POST » ou « GET » légèrement modifiée, dans le fait qu'elle ne contient pas directement le chemin d'accès à la ressource mais l'URL de cette ressource.
Exemple : `GET /index.html HTTP/1.1`
devient : `GET http://mon_serveur_web/index.html HTTP/1.1`

Fonctionnement du proxy pour une connexion TLS

- Le navigateur utilise la méthode `CONNECT` pour demander au proxy d'ouvrir un tunnel :
  « *CONNECT p-fb.net:443 HTTP/1.1* »
- le proxy créé une socket vers le serveur Web demandé ;
- le proxy informe le client que le "tunnel" est ouvert avec une réponse HTTP :
  `HTTP/1.1 200 OK`
- Le proxy transfert les paquets du client vers le serveur et inversement tant qu'aucun des deux ne ferment la socket.

Source : RFC 7231 - 4.3.6 CONNECT

Travail

Écrire un Proxy pour le protocole HTTP qui écoute sur le port 8080 :

- réalise un **filtrage** sur le contenu au format HTML : insertion de texte dans le titre, remplacement de texte, blocage de l'accès au contenu, suppression des ressources au format mp4, *etc*.
- soit **configurable** par l'intermédiaire d'un accès Web capturé par le proxy (ce contenu est fourni par le proxy) :
  à l'entrée d'une URL particulière dans le navigateur, on accède dans son navigateur à une interface en HTML pour configurer le proxy : (dés)activation de filtrage, édition des mots interdits, *etc*.
- gère la **requête au format « `GET` »** en provenance du navigateur ;
- gère la **requête au format « `POST` »** (les données du POST en provenance du navigateur sont à faire suivre au serveur auquel se connecte le proxy) ;
- gère également les **connexions TLS**.
  Dans votre rapport vous pourrez donner le(s) méthode(s) de filtrage ne pouvant pas être appliquée(s), pourquoi et comment faire pour quelle(s) soi(en)t applicable(s)

Votre programme de proxy mémorisera les **paramètres** dans des fichiers de configuration au format texte.

Remarques : pour le bon fonctionnement du proxy, il est nécessaire de :

- dans la requête transmise au serveur et en provenance du navigateur :
  - supprimer les lignes commençant par : `Connection: Keep-Alive` et `Proxy-Connection: Keep-Alive` ;
  - supprimer la ligne commençant par `Accept-Encoding: gzip` ;
- vous assurer de faire des **requêtes** en version `HTTP/1.0` ;
- pour la **configuration**, vous pouvez envoyer, par exemple, un formulaire contenant un `TEXTAREA` contenant la liste des mots à filtrer et proposer à l'utilisateur de les modifier.

Examples

Après avoir configuré Firefox pour utiliser un proxy pour **http** et **https** :

▷ pour une connexion HTTP :

```
GET http://p-fb.net/ HTTP/1.1
Host: p-fb.net
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_2; rv:112.0esr) Ge
cko/20000101 Firefox/112.0esr
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/jxl,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

▷ pour une connexion TLS

```
CONNECT p-fb.net:443 HTTP/1.1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:102.0) Ge
cko/20100101 Firefox/102.0
Proxy-Connection: keep-alive
Connection: keep-alive
Host: p-fb.net:443
```

La commande est `CONNECT` *au lieu de* `GET`.